Tesis por compendio

Agente para Recuperación Automática de Información en diversos entornos basado en técnicas de Inteligencia Computacional

Autor:

Álvaro Ariel Gómez Gutiérrez

Directores:

Carlos León de Mora

Joaquín Luque Rodríguez

Departamento de Tecnología Electrónica, Universidad de Sevilla.

C/ Virgen de África, 7. Sevilla 41011 (España)

Tlfno.: (+34) 954 55 12 13, Fax: (+34) 954 55 28 33.

ariel@us.es

Agradecimientos

Me gustaría agradecer a mis directores de tesis, Dr. Carlos León de Mora y Dr. Joaquín Luque Rodríguez el impulso y confianza mostrados a lo largo del desarrollo de la investigación y la redacción de esta tesis así como sus claras indicaciones.

También me gustaría agradecer el apoyo recibido de mis compañeros, especialmente el de todos aquellos que se brindaron a ayudar en las tareas que fueran necesarias (y a los que nada pude delegar).

A mis padres y a mi hermana por su fe en mi capacidad de la que nunca dudaron.

Y especialmente a mi familia por permitirme disponer de todas las horas de mi tiempo que les correspondían y yo usé para realizar esta tesis.

Gracias a todos.

Resumen

Esta tesis presenta algunos de los diferentes trabajos de aplicación en los que he participado para resolver problemas reales de recuperación de información en diversos entornos.

Dichos trabajos se han realizado en colaboración con distintas empresas, donde se han implementado diversos prototipos de agentes inteligentes aplicando las técnicas de inteligencia computacional que sustentan los métodos de recuperación de información desarrollados, con la finalidad de crear agentes inteligentes para resolución de problemas en los que se necesita realizar una recuperación de información.

El método de recuperación de información desarrollado da lugar a las publicaciones que se adjuntan en el capítulo 8 de esta tesis; y su aplicación propicia una entrada en la oficina de registro de la propiedad intelectual y la implementación del Agente Inteligente de ayuda a la navegación por el portal web de la Universidad de Sevilla.

Índice de contenido

Índice de figuras

Índice de tablas

Abreviaturas

- **AKS**: (*Accumulated Knowledge System*). Conjunto de Conocimiento Acumulado.
- **ANN**: (*Artificial Neural Network*) Redes neuronales artificiales.
- **API**: (*Application Programming Interface*). Interfaz de aplicaciones de programación.
- **AV**: Avatar.
- **CCA**: Conjunto de Conocimiento Acumulado.
- **CDTI**: Centro para el Desarrollo Tecnológico de Andalucía.
- **CI**: (*Computational Intelligence*). Inteligencia computacional.
- **CTA**: Corporación Tecnológica de Andalucía.
- **EEES**: Espacio Europeo de Educación Superior.
- **FE**: (*Fuzzy Engine*). Motor de lógica difusa o motor difuso.
- **FL**: (*Fuzzy Logic*). Lógica difusa.
- **GUI**: (*Graphical User Interface*). Interfaz gráfica de usuario.
- **IC**: Ingeniero de Conocimiento.
- **IPA**: (*Intelligent Pedagogical Agent*). Agente inteligente pedagógico.
- **IR**: (*Information Retrieval*). Recuperación de información.
- **KNN**: (*k-Nearest Neighbors*). K-vecinos más próximos.
- **LMS**: (*Learning Management System*). Plataforma de enseñanza virtual.
- **LN**: Lenguaje Natural.
- **LSL**: (*Linden Scripting Language*). Lenguaje de programación de script desarrollado por Linden research labs.
- **MAS**: (*Multi-Agent System*). Sistema multi agente.

- **MMORPG**: (*Massively Multiplayer Online Role Playing Game*). Juegos de rol online multijugador.
- **OPI**: Organismo Público de Investigación.
- **PAIDI**: Plan Andaluz de Investigación, Desarrollo, e Innovación.
- **REST**: (*REpresentational State Transfer*). Descripción de interfaz entre sistemas que utilizan http para obtener datos
- **RSS**: (*Really Simple Syndication*). Formato XML para compartir contenido en la web.
- **SCORM**: (*Sharable Content Object Reference Model*). Objetos pedagógicos estructurados.
- **SL**: Second Life.
- **TIC**: Tecnologías de la Información y Comunicación.
- **tf-idf**: (*term frequency – inverse document frequency*). Método de asignación de pesos a los términos índices basado en la frecuencia de aparición del término en su documento y en el resto de la colección.
- **TV**: Tutor Virtual.
- **TW**: (*Term Weight*). Peso asignado al término índice.
- **VSM**: (*Vector Space Model*). Modelo de Espacio Vectorial. Modelo en el que se representan los documentos de una colección mediante vectores en un espacio n-dimensional.
- **XML**: (*eXtensible Markup Language*). Lenguaje de definición de datos para intercambio de información estructurada.

Palabras clave

- *Recuperación de información, inteligencia computacional, lógica difusa, lógica borrosa, vector space model, conjunto de conocimiento acumulado, agente inteligente, sistema multiagente*.

1 Introducción y Objetivos

1 Introducción y Objetivos

1.1 Introducción

La presente tesis se enmarca en la problemática de la recuperación de información. En particular, expone un novedoso sistema, basado en lógica difusa, para la implementación de agentes inteligentes en diversos ámbitos.

Por recuperación de información se entiende buscar, de forma automática, todos los documentos, dentro de una colección de documentos diversos relacionados, con un cierto grado de relevancia, con una consulta formulada por un usuario. La relevancia del documento vendrá dada por la afinidad del mismo a unos parámetros especificados. Además, se trata de afinar el proceso para que el número de documentos devueltos sea mínimo y corresponda sólo a los similares a la consulta.

En ocasiones, la consulta puede consistir en uno de los documentos pertenecientes a la propia colección con la intención de recuperar todos los otros documentos presentes en la colección similares al señalado como muestra.

El objetivo de esta tesis es el desarrollo y la aplicación de agentes inteligentes basados en lógica difusa para recuperación de información heterogénea en ámbitos

diferentes a los portales web. En concreto, la aplicación de estos agentes se realiza en el ámbito de la ayuda a la docencia (proyecto Aulas Virtuales), y en el ámbito de los asistentes virtuales para simplificar el uso de aplicaciones de usuario (proyecto Asistente SmartCity.Eco)

Los agentes inteligentes desarrollados utilizan el método de recuperación de información, el método de asignación de pesos, y la estructura de almacenamiento de información desarrollada en las publicaciones adjuntas. En dichas publicaciones se justifica el buen funcionamiento de estos métodos, así como la mejora de rendimiento en la recuperación de información contenida en portales web frente al modelo de espacio vectorial (Vector Space Model, VSM) y el método de asignación de pesos tf-idf. El capítulo 4 resume las pruebas realizadas que avalan el rendimiento de los métodos que se utilizan.

1.2 Estructura de la tesis

La presente memoria se estructura como sigue: En el capítulo 2 se hace una revisión de las técnicas de recuperación automática de información que dan lugar al desarrollo del método basado en lógica difusa presentado en las publicaciones adjuntas y se describe en detalle el modelo de espacio vectorial de representación de documentos y el método de asignación de pesos tf-idf. El capítulo 3 está dedicado a la descripción de los desarrollos del método de asignación de pesos basado en lógica difusa y al algoritmo de recuperación de información basado en lógica difusa aplicados en esta tesis. En el capítulo 4 se hace una descripción de los aspectos técnicos más relevantes de las publicaciones que se adjuntan y el capítulo 5 describe los resultados de la aplicación en proyectos de investigación y desarrollo de los sistemas desarrollados objeto de esta tesis. En este capítulo, mediante el análisis de los resultados de aplicación se muestra con detalle el alcance de la investigación realizada. Por último, el capítulo 6 recopila las conclusiones alcanzadas.

1.3 Resumen global de los resultados

En los últimos años he participado como autor en los dos artículos en revistas internacionales, y un capítulo de libro, donde se han expuesto los resultados obtenidos con la investigación realizada sobre Recuperación de Información basada en métodos de inteligencia computacional y la aplicación en Agentes para realizarla.

En este apartado se resumen las aportaciones más relevantes realizadas durante este tiempo, cuyo texto completo se incluye en el capítulo 8 de la presente memoria de tesis.

Publicaciones en revistas internacionales en las que soy coautor:

- **A1:** *A FUZZY LOGIC INTELLIGENT AGENT FOR INFORMATION EXTRACTION: INTRODUCING A NEW FUZZY LOGIC-BASED TERM WEIGHTING SCHEME*
    - **Autores:** *Jorge Ropero, Ariel Gómez, Alejandro Carrasco, y Carlos León.*
    - **DOI:**10.1016/j.eswa.2011.10.009
    - **Publicación:** Expert Systems with Applications, Volume 39, Issue 4, March 2012, Pages 4567-4581.
    - **Breve descripción:** En esta publicación se definen los parámetros de un nuevo método de recuperación de información al que se accede mediante Lenguaje Natural. También se define un método de asignación de pesos a los términos-índices. El núcleo de ambos métodos es un motor de lógica difusa. Como prueba de funcionamiento, se realizan pruebas sobre la recuperación de información aplicada a los contenidos del Portal web de la Universidad de Sevilla. Las pruebas realizadas demuestran que los portales web constituyen un ámbito de aplicación en el que el método de recuperación de información desarrollado, basado en lógica difusa, ofrece muy buenos resultados.
    - **Índice de calidad:** La publicación está indexada en el JCR con índice de impacto de 1.854 (Q1) y el artículo ha recibido 6 citas.
    - **Mi contribución:** Desarrollo del algoritmo de IR, implementación del motor difuso para IR mediante Unfuzzy, determinación de reglas del motor difuso para IR y TW, determinación de los estimadores de calidad (métricas), diseño de pruebas de funcionamiento, análisis de resultados, optimización del algoritmo de IR, colaboración en la redacción del artículo.

- **A2:** *SABIO: SOFT AGENT FOR EXTENDED INFORMATION RETRIEVAL*
    - **Autores:** *Ariel Gómez, Jorge Ropero, Alejandro Carrasco, Carlos León, y Joaquín Luque.*
    - **DOI:** 10.1080/08839514.2013.774204
    - **Publicación:** Applied Artificial Intelligence, Volume 27, Issue 4, 1 April 2013, Pages 249-277.
    - **Breve descripción:** En esta publicación se describe la implementación de un Agente Inteligente para recuperación de información utilizando los métodos desarrollados y detallados en publicaciones anteriores. Se programa el motor difuso, sus reglas, y los algoritmos de recuperación de información, utilizando el entorno Borland C++ Builder. Se implementa una base de datos para almacenar el conocimiento del Agente y sus parámetros mediante Access. Se diseñan pruebas de funcionamiento y se optimiza el algoritmo de recuperación mediante análisis de los resultados. El resultado de las pruebas señala que la implementación de un Agente Inteligente basado en los métodos desarrollados es viable y alcanza todos los objetivos inicialmente propuestos. De forma general también se comprueba que la implementación del motor difuso realizada mantiene la funcionalidad del configurado en el desarrollo del método y que los coeficientes de peso asignados mediante el Agente son muy similares a los propuestos por el Ingeniero de Conocimiento.
    - **Índice de calidad:** La publicación está indexada en el JCR con índice de impacto de 0.402 (Q4) y el artículo ha recibido 1 cita.
    - **Mi contribución:** Desarrollo de varios motores difusos en Unfuzzy, implementación del motor difuso genérico en C, definición e implementación de las reglas para el motor difuso de IR y TW, implementación del Agente Inteligente para IR, optimización del algoritmo de IR programado,

implementación del Agente asignador de coeficientes de peso, diseño de la estructura e implementación de una Base de Datos para almacenamiento del CCA, diseño de pruebas del Agente de IR y del Agente asignador de pesos, análisis de resultados de las pruebas, redacción del artículo.

Capítulos de libro en los que soy coautor:

- **L1:** *TERM WEIGHTING FOR INFORMATION RETRIEVAL USING FUZZY LOGIC.*
    - **Autores:** *Jorge Ropero, Ariel Gómez, Alejandro Carrasco, Carlos León, y Joaquín Luque.*
    - **DOI:**10.5772/37837
    - **Publicación:** Fuzzy Logic - Algorithms, Techniques and Implementations, Prof. Elmer Dadios (Ed.), ISBN: 978-953-51-0393-6, InTech, DOI: 10.5772/2663
    - **Breve descripción:** En este capítulo de libro se realiza un estudio del rendimiento de un proceso de recuperación de información mediante el diseño de unas pruebas y la comparación de los resultados obtenidos aplicando un método de asignación de pesos basado en lógica difusa desarrollado por los autores, y el método clásico tf-idf. En el análisis se categorizan los resultados bajo dos criterios novedosos. El análisis de resultados constata que el uso de lógica difusa en los algoritmos de asignación de pesos y de recuperación de información mejora los resultados obtenidos respecto al uso del método clásico de tf-idf.
    - **Índice de calidad:** Este libro está citado 4 veces en su conjunto y el capítulo 1 vez más.
    - **Mi contribución:** Diseño de las pruebas, categorización de preguntas-tipo, análisis de resultados y discusión de resultados, colaboración en la redacción del capítulo.

El contenido de estos artículos y el capítulo de libro se describe en los capítulos 3 y 4 de esta tesis y su texto completo se adjunta en el capítulo 8.

Además, la investigación realizada también ha dado lugar al expediente SE-984-11 con número de entrada 201199901248860 en el Registro de la Propiedad Intelectual de Andalucía, y al Agente Inteligente de ayuda a la navegación por el portal web de la Universidad de Sevilla.

En el capítulo 5 de esta tesis se describen otros resultados de aplicación que aún se encuentran en fase de pre-comercialización: agente inteligente tutor virtual, y agente recomendador de servicios de la plataforma SmartCity.ECO.

2 Técnicas de Recuperación Automática de Información

2 Técnicas de Recuperación Automática de Información

La problemática de la recuperación de información (Information Retrieval, IR) comienza a despertar interés desde la década de los 50 (Luhn, 1953), (Lesk, 1969), (Blair, 1979), (Croft, 1986), (Belew, 1989), (Liu, et al., 2001), (Zhao & Karypis, 2002) y se ha aplicado ampliamente a documentos de texto; pero es con el auge de la información disponible en Internet y en las grandes bases de datos de documentación cuando su interés crece y se impulsa su desarrollo.

Esta amplia oferta de información y fácil acceso a la misma, que constituye una enorme ventaja para los buscadores de información, también actúa frecuentemente en contra de los intereses del usuario debido a que se hace muy difícil seleccionar los contenidos realmente interesantes de entre toda la información disponible. Esto ocurre, en muchos casos, simplemente por cuestión del volumen de la información a revisar.

Para dar respuesta a esta necesidad, en el campo de las colecciones de documentos de texto, surgen varios métodos para implementar la recuperación automatizada de la información. Algunas de las propuestas más desarrolladas son:

el modelo de espacio vectorial (Vector Space Model, ó VSM) (Wu & Salton, 1981), (Salton, et al., 1983), (Yu, et al., 1983), (Wong, et al., 1985), (Croft, 1984), (Raghavan & Wong, 1986), (Salton & Buckley, 1988), (Salton, 1989), (Lee, et al., 1997), (Paijmans, 1999) el método los k-vecinos más próximos (k-Nearest Neighbors, ó KNN) (Laha, 2007), los modelos de clasificación Bayesianos (Ruiz & Srinivasan, 2002), (Lu, et al., 2002) y las redes neuronales (Artificial Neural Network, ANN) (Cummins & O'Riordan, 2006) (Thompson & Croft, 1985).

Desde el inicio del estudio de la recuperación automática de información se consideraba que ciertas palabras extraídas de un documento son capaces de identificar, o representar, el significado del texto del que se extraen siendo consideradas como los representantes del documento o de la consulta del usuario. También se sugería que el proceso de recuperación automática de textos podría diseñarse basándose en la comparación de los identificadores presentes en los documentos de una colección y los que aparecen en la consulta del usuario.

Otra alternativa para la representación de un documento consiste en realizar una asignación manual de términos (indexación del documento), presentes o no en el mismo, por una persona con conocimiento sobre el tema. En general, esta persona es conocida como un Ingeniero de Conocimiento (IC).

De entre todas las técnicas de IR mencionadas, el denominado Modelo de Espacio Vectorial (VSM) es el utilizado con mayor frecuencia por su simplicidad y su alta velocidad de procesado.

Debido a este extendido uso y a que es directamente comparable con el sistema basado en lógica difusa (Fuzzy Logic, FL) desarrollado, se describen a continuación resumidamente sus características.

En este modelo, cada documento se representa por un vector constituido por ciertas palabras, denominadas términos-índice (index terms), contenidas en dicho documento a las que se les asigna un coeficiente, denominado de peso (term weight), relacionado con la importancia de la presencia de dicha palabra en el documento que la contiene. De forma habitual se aplica el método denominado tf-idf para realizar la asignación de dichos coeficientes de peso, siendo mayor el valor asignado cuanto más veces aparezca el término índice en el documento que se está caracterizando (term frequency, tf), y disminuyendo dicho valor según la cantidad de otros documentos en los que también aparece el término en cuestión (inverse document frequency, idf). La semejanza de dos documentos, o de una pregunta y un documento, se determina mediante la distancia entre los vectores que los representan. Para ello es frecuente usar la medida del coseno que forman ambos

vectores en su universo de representación. En el apartado 2.1 se describe este modelo y el método tf-idf en detalle.

En las aplicaciones basadas en VSM cobran importancia conceptos como consultas, perfiles de usuario, clustering, o relaciones jerárquicas (Haase, et al., 2002), (Cordón, et al., 2004), (Mercier & Beigbeder, 2005), (Subasic & Huettner, 2001), (Horng, et al., 2005), (Rios, et al., 2006), (Moradi, et al., 2008), (Zhang & Zhang, 2003).

El método VSM obtiene en general unos resultados bastante aceptables aunque adolece de ignorar algunos aspectos que consideramos de importancia, como se justifica en las publicaciones adjuntas.

El sistema de recuperación de información basado en lógica difusa expuesto en esta tesis parte también del concepto de representación de los objetos mediante unos términos índices presentes en la descripción textual del objeto. A estos términos índices también se les asocia un peso cuyo valor depende directamente de la importancia de la relación del término índice con el objeto en cuya representación aparece. No obstante, como se describe en el capítulo 3, el valor del peso asignado tiene en cuenta los parámetros del método tf-idf y otros dos añadidos, y su valor se obtiene mediante aplicación de unas reglas de lógica difusa.

El método desarrollado mejora los resultados de métodos tradicionales como el VSM sobre todo cuando la información a recuperar es vaga, imprecisa, o la consulta se hace de forma poco rigurosa. Además, ofrece como respuesta no sólo el objeto u objetos que claramente se identifiquen con la consulta recibida sino también objetos relacionados con ella. Esto es muy útil en el caso de que el usuario no sea muy experto en el tema sobre el que solicita información y por ello no incluya elementos claves en su consulta, o incluso que introduzca en ella elementos erróneos.

El método desarrollado también tiene la virtud de ser susceptible de aplicación no sólo a colecciones de documentos sino a todo aquello que pueda ser descrito verbalmente, característica esencial para el objetivo que se persigue en el trabajo desarrollado en esta tesis.

Hay que considerar que el objetivo de cualquier sistema de acceso a información es satisfacer las necesidades de los usuarios que demandan respuesta sobre los recursos accedidos por dicho sistema. No obstante, existen diversos problemas en el uso de estos sistemas de acceso al conocimiento:

- En muchas ocasiones los usuarios no son conscientes de la información que verdaderamente necesitan solicitar para satisfacer su necesidad y formulan sus consultas de forma vaga o imprecisa.

- Realizar consultas en algunos sistemas, tales como las bases de datos, no suele ser una tarea fácil y puede requerir conocimientos que cualquier usuario no posee como sentencias en lenguajes de programación, o uso de una determinada sintaxis en la consulta.

Por tanto, surge la necesidad de buscar metodologías que incorporen la capacidad que tienen las personas de tomar decisiones concretas en entornos imprecisos o afectados de incertidumbre. Dos de las principales metodologías desarrolladas con este propósito son las Redes Neuronales Artificiales y la Lógica Difusa. Este conjunto de metodologías se conoce como Soft Computing o Inteligencia Computacional (Computacional Intelligence, CI).

Los motores de búsqueda en portales web y las técnicas clásicas de recuperación de documentos suelen consistir en búsquedas de palabras clave dentro de su ámbito, generalmente la web. El resultado de estas búsquedas puede consistir en miles de elementos siendo, la mayoría de las veces, muchos de ellos irrelevantes o incluso incorrectos.

En la actualidad existen diversos enfoques para manejar la información en un sistema de recuperación de información. Uno de ellos está basado en el Modelo de Espacio Vectorial y otro está relacionado con los conceptos de ontología y web semántica. La información almacenada en textos o documentos no estructurados puede ser accedida combinando los datos estructurados con relaciones jerárquicas y perfiles de usuario (Abulaish & Dey, 2005), (Quan, et al., 2006), (Zhai, et al., 2008), (Martin & Leon, 2009). La Figura 1 muestra un esquema conceptual de las aplicaciones FL para la recuperación de información.

Estas aplicaciones coinciden en dos aspectos fundamentales con el método desarrollado y aplicado en esta tesis.

- La información a manejar es muy grande.
- Se hace necesaria una estructura jerárquica o de agrupamiento para la clasificación de la información.

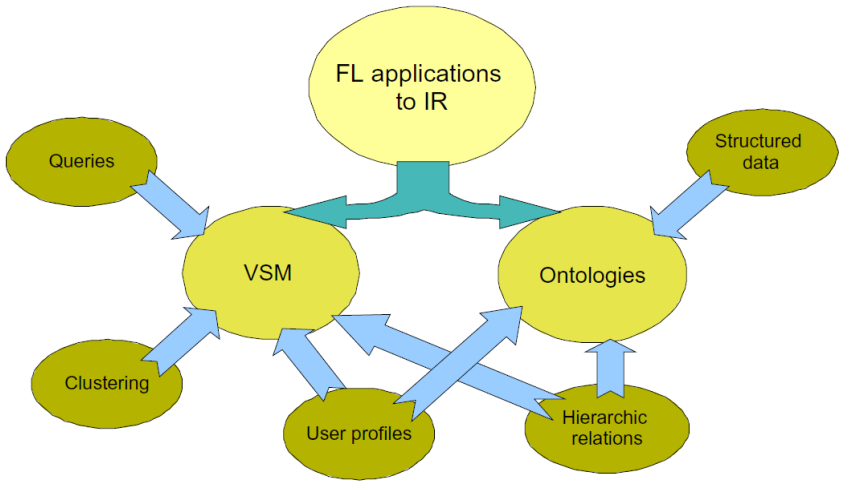


Figura 1: Esquema conceptual de aplicaciones FL para IR

2.1 El Modelo de Espacio Vectorial y el método tf-idf

En esencia, en el modelo VSM, un documento está representado en un espacio multidimensional por un vector, de tal forma que los documentos con significados similares se encuentran cercanos en ese espacio. Por tanto, midiendo la distancia entre los vectores que constituyen la representación de dos documentos se puede inferir la similitud de ambos.

El proceso de aplicación del método a una colección de documentos constaría de tres fases diferentes: en primer lugar, se realizaría una indexación de cada documento mediante la extracción de los términos más relevantes presentes en cada documento; en segundo lugar se asignarían pesos a dichos términos para ponderar la influencia de la presencia de cada uno con el significado del documento en concreto; en tercer lugar se clasificaría la relación del documento con el requerimiento de búsqueda en base a la medida de semejanza de sus vectores.

2.1.1 Indexación del documento mediante los términos más relevantes que contiene

La indexación de un documento consiste en sustituir dicho documento por un vector de términos, contenidos en él, que sean representativos del significado del documento.

$$D = (t_1, t_2, ..., t_i, ..., t_p)$$

Donde D es el vector que representa al documento y ti el término i-ésimo seleccionado de entre su contenido.

En la indexación automática se suele seguir el criterio de señalar como relevantes aquellos términos con una frecuencia de aparición alta así como los de frecuencia de aparición baja. No obstante, la determinación automática de los términos relevantes de un documento no es una tarea trivial, debido a que muchas de las palabras contenidas en el documento no describen su contenido (determinantes, artículos, conjunciones, etc). Además, este tipo de palabras suele aparecer con una frecuencia elevada por lo que, según los criterios mencionados, serían marcadas como relevantes.

Para evitar este problema, se establecen conjuntos de palabras a excluir, denominadas stop-words, de forma que el análisis automático no las considere. Aún así, dependiendo del campo al que pertenezca el documento existen determinadas palabras que podrían falsear la indexación. Si se conoce el ámbito del documento, es posible utilizar tesauros para determinar palabras con significados similares específicos a aquellos del ámbito del documento (Croft & Harper, 1979), (Larsen & Yager, 1993).

No obstante lo anterior, esta indexación automática del documento entraría bastante en el campo del tratamiento del Lenguaje Natural y podría utilizar técnicas de minería de datos, análisis sintáctico, enraizado de palabras, términos compuestos, sinonimia, polisemia, y otras técnicas de análisis y comprensión de textos (Lesk, 1969), (Larsen & Yager, 1993), (Quan, et al., 2006).

Otra opción es una indexación parcialmente manual en la que un Ingeniero de Conocimiento supervise la indexación automática y modifique los términos seleccionados si fuera necesario.

Para que los vectores sean posteriormente comparables, todos tienen que tener la misma longitud, esto es, que deben pertenecer a un mismo espacio multidimensional. Para ello, primero es necesario definir dicho espacio

multidimensional. Una vez indexados todos los documentos que componen la colección, se obtiene un conjunto determinado de términos índices, sean éstos t términos. Cada uno de estos términos constituirá una dimensión del espacio de representación. Una vez definida la secuencia de estos t autovectores, los vectores representativos de los documentos de la colección son reescritos en el formato correcto pasando a tener todos un tamaño homogéneo de dimensión t.

$$D = (t_1, t_2, ..., t_i, ..., t_t)$$

2.1.2 Asignación de pesos a los términos índices. Método tf-idf

Una vez seleccionadas las palabras que constituirán los términos índice que representan al documento es necesario asociarles un peso para indicar en que medida es determinante su aparición en el texto.

Uno de los principales retos en la problemática de la IR es el proceso de asignar a cada término índice un peso que refleje la importancia de la presencia de ese término índice en la identificación del objeto en cuya representación aparece.

En principio, habría dos posibles formas de asignar a cada término índice un valor que represente la importancia de su presencia en la identificación del objeto:

- Un experto en la materia evalúa intuitivamente dicha importancia. Este método es simple pero tiene el inconveniente de depender exclusivamente del Ingeniero de Conocimiento que realice la asignación y no puede ser automatizado.
- Generación de unos pesos mediante un conjunto de reglas.

Dado que esta tarea es grande si el Conjunto de Conocimiento Acumulado (CCA), en este caso la colección de documentos, es grande y que además habría que recalcular si se introducen nuevos documentos donde aparecieran términos índices diferentes a los ya existentes, sólo se contempla la segunda forma ya que la primera no es susceptible de ser automatizada.

En un principio, el reparto de pesos era realizado de forma puramente booleana, si el término aparecía en el vector se le asignaba un valor de 1 y si no aparecía el valor era 0.

Aunque la idea de construir sistemas de recuperación automática de texto basados en los contenidos del texto e identificadores asociados ya surgió a finales de 1950 (Luhn, 1953), fue a finales de 1970 y en la siguiente década cuando Gerald

Salton sentó las bases de la relación existente entre los identificadores y los textos a los que representan (Salton, et al., 1983), (Salton & Buckley, 1988), (Salton, 1989).

Salton sugirió que cada documento D podría ser identificado por vectores de términos t_k y un juego de pesos w_{dk}, que representen el peso, es decir su importancia, del término t_k en el documento D. Así pues, el proceso de identificación se mejoró realizando una asignación de pesos de forma continua con valores entre 0 y 1 correspondiendo los valores más próximos a 1 a los términos más indicativos del significado del documento y los valores más próximos a 0 a aquellos menos relevantes.

Clásicamente, la representación más usada utiliza dos estimadores y es denominada tf-idf. En ella el peso del término (w_i) se relaciona con la frecuencia de ocurrencia de dicho término (t_i) en el documento (term frequency, ó tf), y con la frecuencia inversa de aparición del término en los documentos de la colección (inverse document frequency, ó idf).

El fundamento de estos estimadores tiene su origen en que cuando consideramos un documento de forma individual, un término que aparezca en él con una frecuencia elevada suele señalar al documento. Esto se conoce como term frequency o tf. Por ello, los términos con mayor tf deberían tener más peso que los otros.

Por otro lado, cuando consideramos un conjunto de documentos, la aparición de un mismo término en muchos de ellos lo devalúa para realizar una discriminación, luego cuanto menos documentos lo contengan, mayor es la importancia del término en la identificación del documento que lo contiene y, por ello, el peso que debería asignársele. Esto se conoce como inverse document frequency (idf). El valor del parámetro idf de un término varía inversamente según el número n de documentos, de una colección de N documentos, que lo contienen en su representación.

La expresión utilizada para calcular el coeficiente w_i correspondiente al término t_i del vector D que representa al documento d es:

$$w_i = tf \cdot idf_i = tf(t_i, D) \cdot idf(t_i)$$

Donde $tf(t_i,D)$ es la frecuencia de aparición del término i-ésimo del vector D en el documento d; e idf representa un valor asociado al nº de documentos en cuyos índices aparece el término i-ésimo y que se calcula como:

$$\text{idf}(t_i) = \log\left(\frac{N}{DF(t_i)}\right)$$

siendo N el número total de documentos de la colección, y DF(t_i) el número de documentos de la colección que contienen el término t_i.

Una vez obtenido el coeficiente de peso, el vector representativo del documento es:

$$D = (t_1, w_1; t_2, w_2; ...; t_i, w_i; ...; t_t, w_t)$$

Una vez escritos todos los vectores en el espacio de dimensión t, ya no son necesarios los términos t_i permaneciendo sólo los pesos w_i.

Para compensar el efecto de que un documento más extenso asigne unos pesos mayores que otro más pequeño, se hace necesaria la inclusión de una modificación en los coeficientes. Para ello se introduce un factor de normalización que corrige esta disimetría balanceando los pesos, teniendo en cuenta los valores de los coeficientes del vector índice y haciendo que el módulo de todos los vectores tenga el mismo tamaño, la unidad.

La expresión normalizada de estos términos es la siguiente:

$$w'_{dk} = \frac{w_{dk}}{\sqrt{\sum_{vector} \left(w_{di}\right)^2}}$$

donde w'_{dk} es el peso normalizado del término k-ésimo del índice del documento d.

2.1.3 Determinación del grado de similitud entre el modelo y cada documento

Cuando se recibe una solicitud, se sigue un proceso similar de indexación de la misma como en el caso de los documentos construyendo un vector Q formado por los términos índice que definen la consulta dentro del espacio multidimensional del CCA. El vector resultante tendría la siguiente forma:

$$Q = (q_1, q_2, ..., q_i, ..., q_t)$$

Al igual que en el caso de los documentos, el vector de la consulta tiene que pertenecer al mismo espacio t-dimensional que los documentos indexados. El valor

del vector de consulta es en principio: si un autovector del espacio aparece como término índice de la consulta, su valor es 1, y si no aparece entre los términos de la consulta se le asigna un valor de 0.

Al igual que para los vectores de los documentos una asignación de pesos continua mejora los resultados del proceso de identificación. En este caso, la asignación se realiza de forma estadística. Una expresión habitualmente usada para ello es (Salton & Buckley, 1988):

$$w'_{qk} = \left(0.5 + \frac{0.5 \cdot t_{fk}}{\max t_f} \right)$$

El IC también podría realizar la asignación manualmente. En cualquier caso, se obtiene así un vector de t términos que pueden tomar valores de forma continua entre 0.5 y 1 para los términos del espacio multidimensional presentes en la consulta y de 0 para los que no aparezcan en ella.

$$Q = (q_1, q_2, ..., q_i, ..., q_t)$$

Otra problemática relacionada es la de agrupar los documentos de una colección por similitud de contenidos. En este caso, la consulta corresponde al vector índice del documento que se toma como modelo.

Cuando los vectores D y Q tomaban valores discretos 0 ó 1, la función que evaluaba la similitud entre la consulta y el documento procedía a multiplicar los vectores D y Q correspondiendo su resultado al número de términos que aparecían en ambas representaciones. Con esta asignación de pesos lo que se utiliza para señalar la similitud de un documento con otro es, en definitiva, el número de palabras comunes.

$$Similitud(D,Q) = \sum_{k=1}^{t} w_{qk} \cdot w_{dk}$$

No obstante, cuando los términos pueden tomar cualquier valor entre 0 y 1 de forma continua, la función de uso más extendido para representar la similitud es el coseno del ángulo entre ambos vectores. Teniendo en cuenta la normalización de coeficientes, dicha función adopta la siguiente expresión:

$$Similitud(D,Q) = \frac{\sum_{k=1}^{t} w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2 \cdot \sum_{k=1}^{t} (w_{dk})^2}}$$

De esta forma, se obtiene un valor relacionado con la similitud de significado entre la consulta realizada y cada uno de los documentos de la colección más acorde con el significado del contenido de los documentos comparados.

Definiendo un valor umbral mínimo de similitud para la aceptación de documentos es posible recuperar sólo aquellos relacionados con la consulta recibida lo que, en definitiva, constituye el objetivo perseguido con la IR.

2.1.4 Estimadores de la eficacia del método de Recuperación de Información

En esta problemática se definen clásicamente dos parámetros fundamentales para determinar la eficacia del método empleado: recall (memoria) y precision (precisión) (Salton & Buckley, 1988).

El parámetro recall se define como el número de documentos relevantes recuperados dividido entre el número total de documentos relevantes existentes en la colección.

$$recall = \frac{número\,de\,documentos\,relevantes\,recuperados}{número\,de\,documentos\,relevantes\,en\,la\,colección}$$

El parámetro precision se define como el número de objetos recuperados que realmente son relevantes dividido entre todos los documentos recuperados.

$$precision = \frac{número\,de\,documentos\,relevantes\,recuperados}{número\,total\,de\,documentos\,recuperados}$$

En la extensa bibliografía existente, se proponen otras expresiones para el cálculo de los coeficientes tf-idf buscando una mejora en el recall y precision (Salton & Buckley, 1988), (Paijmans, 1999). Por regla general, son modificaciones sobre las enunciadas y si se mejora un parámetro se suele empeorar el otro.

En el ámbito de estas publicaciones, la pregunta del usuario es un documento modelo y lo que busca es recuperar todos los existentes con temática similar. También se podría construir una pregunta manualmente, incluyendo términos y

asignándoles pesos a mano. Esto supondría que el usuario tiene un conocimiento extenso del sistema y de su funcionamiento.

3 Desarrollos de investigación en Recuperación de Información basados en Lógica Difusa aplicados en esta tesis

3 Desarrollos de investigación en Recuperación de Información basados en Lógica Difusa aplicados en esta tesis

El método tf-idf ofrece unos resultados que funcionan razonablemente bien pero, desde nuestro punto de vista, tiene la desventaja de no considerar dos aspectos fundamentales.

El primero de ellos es tomar en consideración el grado de identificación del objeto si sólo se tiene en cuenta el término índice considerado. Si el grado de identificación es alto, este parámetro debe influir fuertemente en el valor final del peso asignado. Cuanto más claramente identifique al objeto, mayor debe ser el valor asignado al peso del término.

El segundo aspecto no considerado en el método tf-idf está relacionado con los términos compuestos. Cuando aparecen conceptos formados por más de una palabra, el método tf-idf considera sus componentes por separado pero no el término compuesto. El peso de cada término por separado tendría un valor inferior al peso del término compuesto. Por ejemplo, el concepto de "Consejo de Gobierno" está formado por dos términos índices y tendría un peso más alto si aparecen ambos en una consulta que los correspondientes a cada uno por separado. El término compuesto puede identificar muy claramente al objeto mientras que sus componentes pueden estar presentes en la representación de muchos otros objetos.

3.1 Método de asignación de pesos basado en Lógica Difusa

En el método de asignación de pesos basado en FL propuesto, estos dos parámetros, junto con un parámetro relacionado con el clásico tf, y otro parámetro relacionado con el clásico idf determinarán el peso asignado al término. Su valor, que vendrá fijado por la salida que devuelve un motor difuso, se obtiene aplicando unas reglas lógicas a los valores proporcionados como entradas. La Figura 2 muestra el esquema.

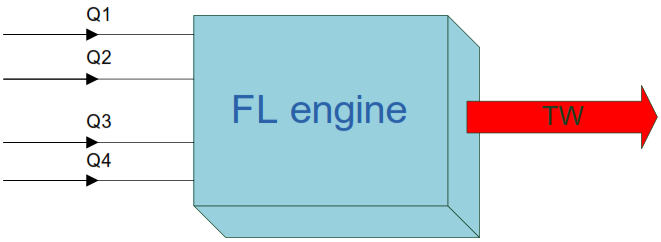


Figura 2: Asignación de pesos. Método basado en FL

Los valores que se usarán como entradas están relacionados con la respuesta a las cuatro preguntas que se formulan a continuación.

- P1.- ¿En cuántos subconjuntos diferentes al propio aparece el término evaluado? Esta pregunta está relacionada con el concepto clásico de idf.
- P2.- ¿Cuántas veces aparece el término evaluado en su subconjunto? Esta pregunta está relacionada con el concepto clásico de tf.
- P3.- El término evaluado, ¿identifica inequívocamente por si mismo al objeto?
- P4.- ¿A cuantas palabras está unido el término evaluado para formar un término compuesto?

3.1.1 Cálculo de los valores de entrada del motor difuso

El rango de valores de todas las entradas está comprendido entre 0.0 y 1.0 por lo que los valores obtenidos como respuesta de las preguntas deben ser normalizados y adaptados al rango.

Para la pregunta P1, si el término aparece muchas veces, el valor asociado debe ser 0 y si no aparece ninguna debe ser 1. Para establecer que se entiende por "muchas veces" y la escala intermedia, se considera el número de subconjuntos de un nivel en los que aparece el término índice y se ordenan de mayor a menor.

El método desarrollado en esta tesis propone tomar el valor de la frecuencia de aparición del término correspondiente al 1% de los términos que aparecen con mayor asiduidad para señalar el valor a partir del cual el coeficiente debe valer 0. Teniendo en cuenta que en el caso de aplicación detallado en el apartado 4.1 se definieron 1114 términos índices, será la frecuencia del undécimo término de la tabla anterior la que debe ser considerada el valor frontera. En el caso correspondiente al análisis realizado, el valor es de 12 apariciones.

En el apartado 4.1.9 se detalla el universo de discurso y la clasificación del valor de las entradas del motor difuso utilizado en 3 conjuntos difusos: BAJO, MEDIO, y ALTO.

De esta forma, si el término índice aparece más de 12 veces en otros subgrupos, el valor asignado al parámetro será 0. Si aparece de 0 a 3 veces (3 veces corresponde aproximadamente a 1/nº de conjuntos difusos del rango de las entradas) se considerará que pertenece al grupo ALTO asignándosele valores entre 1 y 0.7. Operando de forma análoga, si el término índice aparece de 10 a 13 o más veces se considerará que pertenece al conjunto BAJO asignándole valores de 0.3 a

0. El resto de valores corresponderán al conjunto MEDIO y se obtienen por regresión lineal de los restantes. La Tabla 5 del capítulo 4 muestra los valores obtenidos en el caso de aplicación utilizado en las pruebas del método desarrollado.

Este proceso debe ser calculado para cada nivel ya que la división en subconjuntos es diferente, lo que dará lugar a escalas distintas. Para el nivel de apartado, hay que considerar el nº de veces que los términos índices aparecen dentro de un tema, hacerlo para todos los temas y volver a considerar el valor frontera que determina el subgrupo correspondiente al 1% de los términos índices del sistema. A continuación se construye la tabla correspondiente.

De igual forma se procede para el nivel de objeto y se obtiene la tabla correspondiente. En el capítulo 4 se desarrolla en detalle un caso de aplicación y se obtienen los valores numéricos de las tablas correspondientes.

Para averiguar el valor asociado a la pregunta P2.- ¿Cuántas veces aparece el término evaluado en su subconjunto? El razonamiento es análogo, debiendo considerar de nuevo la frontera del 1% de los términos aunque con las siguientes particularidades:

- Cuanto más veces aparece el término en el subconjunto considerado, mayor es el valor que debe ser asignado.
- Este término no tiene sentido en el nivel de objeto dado que todos los conjuntos son unitarios y, por tanto, el término sólo aparece una vez.

En el caso correspondiente al análisis realizado en la 1ª publicación adjunta, el valor es de 12 apariciones. En esas condiciones, la Tabla 6 del apartado 4.1.11 muestra el coeficiente asignado a cada valor de aparición.

Para el caso de la pregunta P3.- El término evaluado, ¿identifica inequívocamente por si mismo al objeto? La respuesta es completamente subjetiva y se proponen tres posibles respuestas: “Sí”, “Algo”, y “No”. La Tabla 9 del apartado 4.1.11 muestra los valores asociados.

El único aspecto que no se ha definido es el valor que toma un coeficiente cuando el término aparece varias veces en un mismo subconjunto (tema o apartado) con valores diferentes. Por ejemplo, puede ocurrir que la respuesta a la pregunta P3 sea “Algo” en un caso y “No” en otro. En este caso, el valor asignado se calcularía mediante una media ponderada de los valores individuales.

Por último, en el caso de la pregunta P4.- ¿A cuántas palabras está unido el término evaluado para formar un término compuesto? Se consideran cuatro posibles

respuestas atendiendo al número de términos a los que esté unido: “A ninguna”, A otra”, “A dos”, “A más de dos”. Los valores asociados a cada una de ellas corresponderán a 0.0 si está unida a más de 2 términos índices, 1.0 si no está unido a ningún otro, y un reparto de los posibles valores según el número de conjuntos difusos en los que se clasifique el universo de discurso. Al considerar una división del universo de discurso en 3 conjuntos difusos y ser 4 los posibles casos de términos relacionados, el valor asignado a este parámetro cuando un término índice está asociado a otro es de 0.7, y de 0.3 para el caso de que esté ligado a otros 2.

La Tabla 10 del apartado 4.1.11 muestra los valores señalados. De nuevo, los valores 0.7 y 0.3 son consecuencia de considerar el número de conjuntos difusos en los que se clasifica el universo de discurso.

Los valores de estos 4 parámetros así calculados constituirán las entradas del motor difuso el cual devolverá un valor de peso asociado al término-índice evaluado en su salida.

3.2 Normalización de los objetos del Conjunto de Conocimiento Acumulado

Mientras que el modelo VSM nace enfocado a la recuperación de documentos de texto, el método desarrollado en esta tesis propone un nuevo enfoque basado en inteligencia computacional y que sería aplicable a conjuntos de conocimiento de cualquier tipo, documentos, imágenes, diagnósticos, páginas web, sentencias de código, enlaces a información, etc, capaces de ser descritos en Lenguaje Natural (LN), tratando de emular a una persona conocedora de la materia que interesa al usuario, y a la que se le puede preguntar en Lenguaje Natural.

La generalidad del método deriva de que al inicio del proceso, los objetos físicos serán sustituidos por sus representaciones en Lenguaje Natural. Mediante este proceso, el sistema ve cada objeto de su Conjunto de Conocimiento Acumulado (CCA) como un conjunto de palabras con un peso asociado sirviendo por tanto las mismas técnicas descritas sea cual sea la naturaleza de los objetos del CCA. De esta forma, la aplicación del sistema desarrollado no queda restringida a CCA de documentos de texto, como ocurre con otros métodos de recuperación de información. Con el proceso de normalización se trata de realizar dicha sustitución de los objetos físicos de cualquier naturaleza por representaciones en Lenguaje Natural.

3.2.1 Representación en Lenguaje Natural de los objetos. Normalización.

En los métodos de IR clásicos, los objetos de su conocimiento son generalmente colecciones de documentos y sus representaciones se construyen con piezas de los objetos mismos, es decir, las palabras contenidas en ellos. En el método basado en FL desarrollado, los objetos del CCA no son necesariamente de tipo texto. Por lo tanto, los métodos de representación de objetos existentes no son directamente aplicables.

Así como el cerebro humano transforma la información recibida por los sentidos y los almacena de forma permanente en el hipocampo y otras estructuras (Kandel, 2006), (Sato, et al., 2004), utilizando sus propias células y proteínas, el método de IR desarrollado construye la representación de objetos usando elementos del propio CCA. Los elementos utilizados por el sistema son los términos-índice pertenecientes a su vocabulario siendo el objeto representado por un conjunto de tamaño variable de estos términos.

En adelante denominaremos vocabulario al conjunto de términos-índice pertenecientes a la representación de algún objeto del CCA.

La representación del objeto no está completa hasta añadir un coeficiente de peso relacionado con la importancia de cada término-índice presente en su representación.

Como se desprende de lo anterior, es necesario asignar una representación en LN a cada objeto del CCA. Esta asignación la realizará la persona experta en la materia a clasificar a la que ya denominamos anteriormente Ingeniero de Conocimiento. El proceso a seguir consta de 2 pasos.

Paso 1.- El primer paso consiste en formular una sentencia, generalmente interrogativa, en LN cuya respuesta sea el objeto que se quiere representar. También es posible utilizar una oración enunciativa correspondiente a una definición del objeto. Denominaremos a esta sentencia pregunta-tipo. De forma general, se formularán más de una pregunta-tipo asociadas a cada objeto.

El conocimiento del IC respecto al lenguaje específico del campo relacionado con los objetos a clasificar, y respecto al modo en el que solicitan información los no expertos, es importante. Cuanto mayor sea su conocimiento, mayor será la fiabilidad de las preguntas-tipo propuestas para la representación de ese objeto porque serán más parecidas a las consultas reales que harán los usuarios para recuperar la información.

Paso 2.- El segundo paso consiste en seleccionar, de entre las palabras que componen la pregunta-tipo, aquellas cuyo significado se relaciona fuertemente con el objeto al que representa. Cada una de estas palabras seleccionadas constituye un término-índice Este concepto guarda un fuerte paralelismo con los términos índices del modelo VSM.

El grupo completo de términos-índice extraídos de las preguntas-tipo enunciadas será la representación del objeto a nivel de procesamiento de LN. Este paso se relaciona con la confección del vector D del método VSM que representaba a un documento de la colección aunque con las siguientes dos diferencias esenciales:

- El vector de representación en el método desarrollado no tiene dimensión t constante como en el modelo VSM y está compuesto por tuplas [a,b] donde "a" es un término-índice y "b" su peso asignado.
- Los términos-índices que lo forman provienen de la unión de términos-índices extraídos de las múltiples preguntas-tipo asociadas al objeto, en caso de que hubiera varias.

Experimentalmente se observa que el cardinal de los grupos de términos-índice extraídos de una pregunta-tipo que representa a un objeto viene a estar comprendido entre dos y seis elementos, esto es, contribuirá a la representación del objeto con entre dos y seis términos-índice.

Con este proceso se realiza una normalización de la naturaleza de los objetos del conjunto de conocimiento pasando de un conjunto de objetos de cualquier naturaleza física a un conjunto de grupos de términos-índice que los representan ante el sistema. En la Figura 3 se puede apreciar el concepto de normalización de un objeto

Cada objeto físico del CCA es único y diferente de los demás pero, al sustituirlo por un grupo de términos-índice, algunos de ellos (o incluso todos) aparecerán en las representaciones (grupos de términos-índice) en LN de otros objetos haciendo que existan múltiples apariciones del mismo término-índice en el CCA.

En el caso del ejemplo, existen más cuadros de Salvador Dalí en los que aparecen relojes blandos (“Reloj blando en el momento de su primera explosión”, u “Osificación prematura de una estación ferroviaria”) por lo que en sus respectivas representaciones en LN aparecerán muchas de las palabras-clave de este objeto.

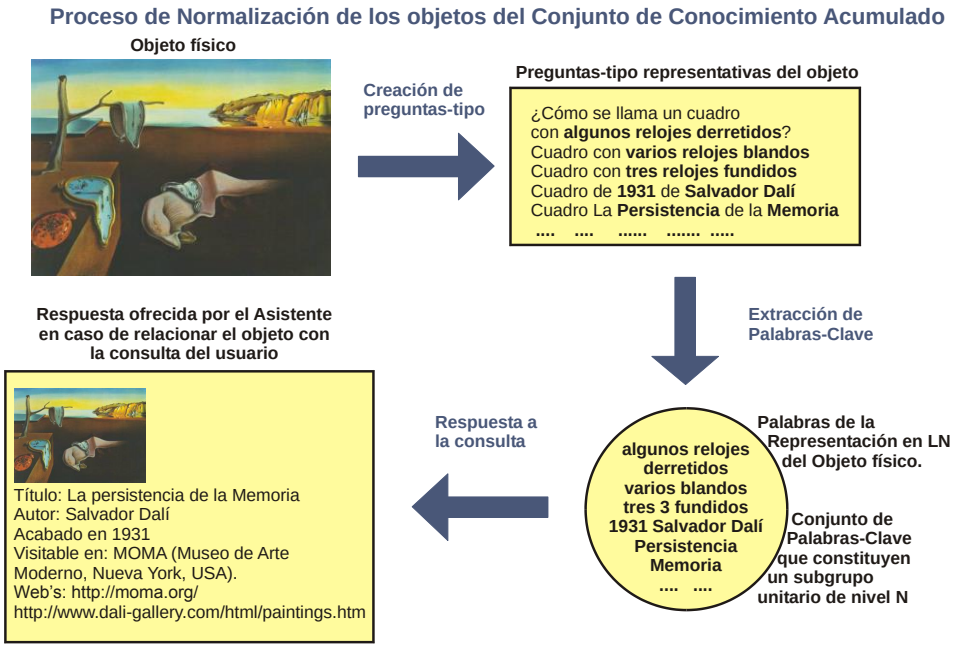


Figura 3: Proceso de normalización de objetos

Además, el grado de relación del significado del término-índice y el objeto en cuyo grupo de términos-índice aparece no tiene por qué ser el mismo en todas las representaciones de los objetos por lo que el valor del término "b" de la tupla sería diferente. Por ejemplo, en el caso del CCA elegido en las publicaciones del capítulo 4 para realizar las pruebas del método propuesto, el término "virtual" es fundamental para determinar que la consulta realizada corresponde al subgrupo de nivel 1 denominado "Universidad Virtual", mientras que no aporta prácticamente información para distinguir un objeto de otro dentro de dicho subgrupo. Por ello, en la representación del subgrupo de nivel 1, al término "virtual" se le asignará una relación de pertenencia a dicho subgrupo de valor elevado mientras que, al mismo término, se le asignará una relación de pertenencia al nivel de objeto de valor bajo.

De lo anterior se desprende que, en el plano de los objetos físicos, éstos (los objetos) pertenecen de forma absoluta (0 ó 1) a los subconjuntos en los que se clasifica el CCA, por lo que nos encontramos ante una teoría de conjuntos de Cantor; mientras que en el plano de su representación en LN, los términos tienen un grado de relación variable con el objeto al que representan, por lo que en este plano de representación en LN de los objetos nos encontramos ante una teoría de

conjuntos difusos. El valor asociado al grado de pertenencia de un término determinado al conjunto de términos-índice que constituyen la representación en LN del objeto puede variar de forma continua de 0.00 a 1.00.

Por ello, la representación en LN del objeto físico no puede estar constituida únicamente por un conjunto de términos-índice determinados, sino que la formarán el conjunto de tuplas [a,b] donde a es un término-índice y b el coeficiente de peso asociado a ese término-índice respecto al subconjunto del CCA en el que se encuentra.

Llegados a este punto, disponemos de una colección de representaciones en LN de los objetos físicos, consistentes en grupos de términos-índices extraídos de las preguntas-tipo construidas a partir de los objetos físicos, y sus correspondientes coeficientes de peso. Estas representaciones que componen el CCA deben ser clasificadas para su almacenamiento ordenado y posterior extracción.

3.2.2 Clasificación de los objetos del conjunto de conocimiento.

Para determinar con qué objetos se relaciona la consulta realizada, los métodos de IR clásicos realizan la evaluación de similitud para cada uno de los objetos del CCA, recorriendo todos los documentos de su colección. Este proceso es largo y claramente ineficiente. Si partimos de la suposición de que ante una demanda de recuperación de información anteriormente almacenada (extracción de un recuerdo), el pensamiento humano actúa realizando un proceso similar a la aplicación de un filtro paso de banda sobre el conjunto de todos sus recuerdos (todo el Conjunto de Conocimiento Acumulado), eliminando toda aquella información que estime que no guarda un determinado grado de relación suficiente con el recuerdo que se intenta recuperar para concentrar su búsqueda en los recuerdos que considere relacionados, podremos intentar que el método de IR desarrollado emule el procedimiento supuesto.

Este enfoque presupone que la información (el conocimiento del individuo) que se va adquiriendo se guarda de forma ordenada y se clasifica agrupada temáticamente según ciertos parámetros que dependen de lo que resulte significativo a cada individuo para que esas características comunes a los recuerdos y a la consulta permitan eliminar los recuerdos no relacionados con una futura petición de recuerdo.

La propuesta realizada para implementar esta capacidad de una forma eficiente es agrupar los objetos del CCA en subgrupos atendiendo a ciertos criterios que

concretaremos más adelante. De esta forma, la evaluación de la relación de la consulta del usuario con el subgrupo del CCA considerado permitirá descartar múltiples objetos de una vez sin necesidad de compararlos todos.

Por último, la propuesta incluye un refinamiento de la organización para minimizar el número de evaluaciones a realizar y, por consiguiente, el tiempo empleado en la determinación. Para ello, se considera una agrupación iterativa de subconjuntos del CCA formando subdivisiones de manera que cada nueva agrupación contenga más elementos que la anterior. La evaluación se hará comenzando por los subconjuntos que contengan mayor número de elementos para "dejar pasar" menos elementos a etapas sucesivas de procesado descartando así inicialmente el mayor número de objetos posible.

En definitiva, una vez sustituidos los objetos por sus representaciones en LN deberemos clasificarlos (los objetos del CCA) en una estructura de 3 ó 4 niveles correspondiendo el nivel más bajo (de mayor número) a la representación individual de un objeto (al conjunto de términos-índice que lo representa) y los niveles siguientes a grupos de objetos con características similares.

Dado que la extracción de los objetos se realizará mediante requerimientos en LN, la principal característica de agrupación será la aparición de términos-índice iguales o relacionados en las representaciones de los objetos.

En la Figura 4 se pueden apreciar los conceptos de:

- Conjunto unitario de nivel N: esfera que contiene el conjunto de pares ordenados [a,b] correspondientes a las términos-índice y sus coeficientes de peso asociados, que constituyen la representación en LN de un objeto físico.
- Agrupación de objetos en un subgrupo de nivel N-1: cada división del cajón en el que se encuentran varios objetos con características similares.
- Agrupación de subgrupos de nivel N-1 en grupos de nivel N-2: cada cajón en el que se agrupan subgrupos de nivel N-1 con características comunes.
- Agrupación de subconjuntos de nivel N-2 en familias de nivel N-3: cada archivador con una serie de grupos de nivel N-2 con características comunes.

- CCA completo constituido por todas las familias de nivel N-3 al que asignaríamos el nivel 0.

Obsérvese que el número de elementos en los que se divide cada subconjunto no tiene por qué ser el mismo. En la Figura 4 se puede apreciar que la subdivisión del CCA de nivel 1 más alejado (archivador del fondo) está compuesto por un número menor de subdivisiones de nivel 2 (tiene menos cajones) que el más cercano.

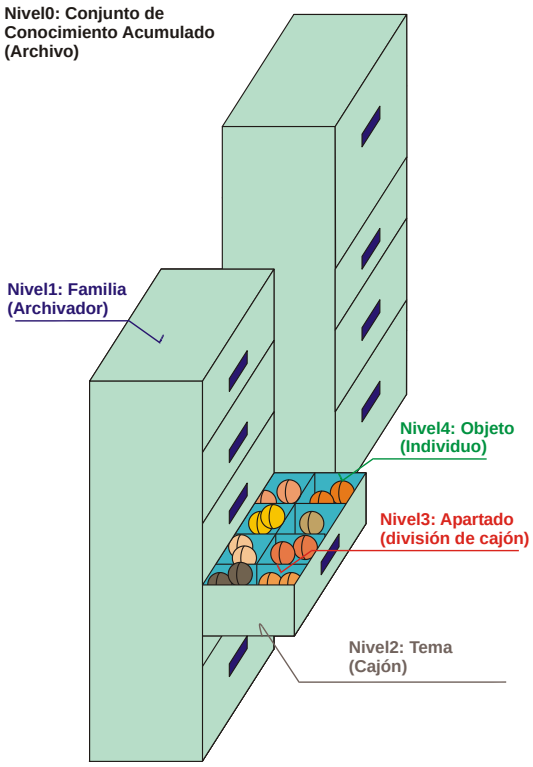


Figura 4: Subdivisiones del CCA

El método desarrollado realiza una clasificación de los objetos de manera completamente vertical, forzando a que cada objeto sólo sea accesible a través de un único camino del árbol de clasificación, es decir, clasificándolo bajo un único criterio (o grupo de criterios).

El que el CCA esté muy verticalizado hace que la búsqueda de un elemento sea muy simple porque su lugar está perfectamente definido así como la/s familia/s

a la/s que pertenece. A la hora de definir criterios de búsqueda, típicamente filtros en BD, es fácil de hacer, mientras que un conjunto de elementos con relaciones muy transversales es de difícil filtrado para encontrar a un individuo concreto.

La decisión del grupo de entre los posibles al que se asigna cada objeto la realiza el Ingeniero del Conocimiento basándose, además de los criterios de temática común, en la similitud del grupo de términos-índice que representa a los objetos que se agrupan juntos y no en criterios físicos del objeto (color, tamaño, material, etc). Naturalmente, los términos-índice presentes en su representación en LN podrían corresponder a características físicas del objeto a clasificar, pero la clasificación se seguiría haciendo en base a los términos-índice y no directamente a las características físicas de los objetos, aunque en ese caso coincidieran.

De esta forma los objetos quedarán clasificados en una estructura arborescente de raíz única en la que se establecerán varios niveles. Cuyos conjuntos del último nivel (Nivel N) serán todos de tipo unitario, como ya se ha enunciado, conteniendo un grupo de palabras-clave que representen a un único objeto del CCA.

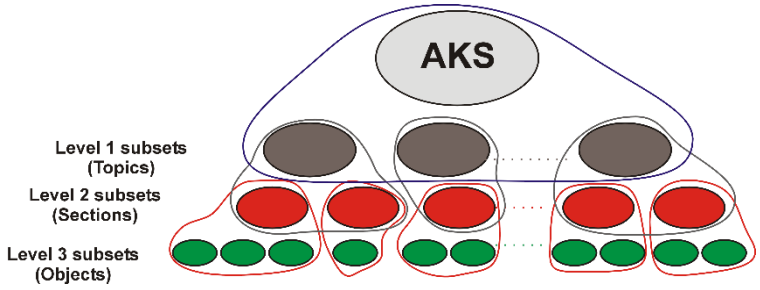


Figura 5: Agrupación de objetos del CCA

Utilizando el modelo seguido en los libros de texto, en los que su contenido es clasificado mediante la estructura de tema y apartado, se asimilan los objetos del CCA a los contenidos de los textos y se agrupan en 3 niveles (Objeto, Apartado y Tema). En sistemas que incorporen contenidos pertenecientes a áreas muy alejadas entre si, como por ejemplo jardinería y electrónica, se establece un nivel adicional de Materia o Familia lo que sería equivalente a distribuir el conocimiento entre diferentes libros. Mediante el análisis de los resultados de las pruebas realizadas, se constata que con la estructura propuesta de tres o cuatro niveles es posible realizar una clasificación suficientemente eficiente para la posterior recuperación de la información.

La clasificación, o agrupación, de los objetos en subgrupos de nivel N-1 se realiza en dos pasadas, en la primera se aplica un método top-down clasificando los objetos del CCA en subdivisiones (conjuntos) según las familias temáticas que hayan aparecido en la fase de representación en LN de los objetos. Una vez realizada la primera clasificación de los objetos, se aplica el criterio de similitud de palabras-clave contenidas en los grupos que definen a los objetos de una misma familia realizando una pasada bottom-up y reubicando aquellos objetos que encajen mejor en otras familias incluso si contravinieran los criterios empleados en la clasificación top-down.

No se debe de perder de vista que el método utilizado por el usuario para la recuperación de la información será una consulta en Lenguaje Natural por lo que es este criterio el que debe predominar. Para los niveles superiores, sólo se realizará el proceso bottom-up utilizando el criterio de similitud de palabras-clave.

Con la clasificación propuesta, el acceso a cualquiera de los elementos tiene una ruta única y de igual longitud que la de los otros.

No se debe de olvidar que esta clasificación sólo constituye la mitad de las representaciones en LN de los conjuntos en los que se agrupan los objetos debiendo añadir posteriormente la información de los coeficientes de peso correspondientes a la pertenencia de los términos-índice a cada conjunto de cada nivel según se describió en el apartado anterior.

3.3 Método de recuperación de Información basado en Lógica Difusa

En el método VSM y otros métodos clásicos la consulta o el documento modelo es comparado con todos los objetos de la colección que constituyen el Conjunto de Conocimiento Acumulado, mientras que el método basado en Lógica Difusa desarrollado sólo compara unos cuantos de los objetos lo cual mejora bastante el rendimiento del proceso de recuperación de la información y disminuye la carga computacional que genera la búsqueda.

Para ello, los objetos deben agruparse en la estructura jerárquica en forma de árbol similar a la que adoptaría una ontología propuesta en el apartado anterior. Esta estructura consta de varios niveles de forma que cada subgrupo de un nivel contiene uno o varios subgrupos del nivel inferior. La Figura 6 muestra la estructura propuesta.

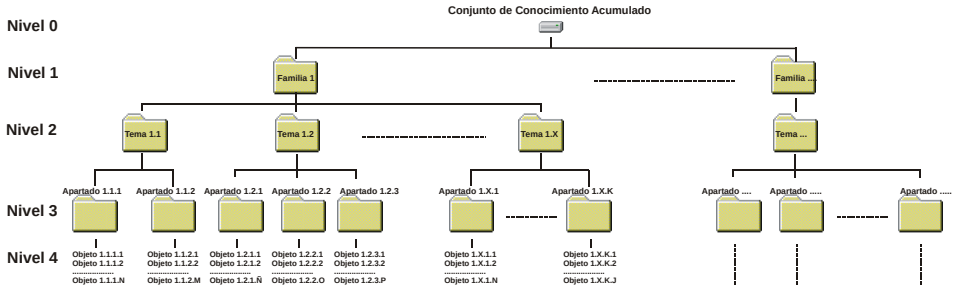


Figura 6: Estructura del CCA

3.3.1 Normalización de la consulta recibida.

La interfaz de usuario del método desarrollado es el Lenguaje Natural. Por ello, la consulta de usuario será un texto en LN en el que se demande la información deseada como si se le estuviera pidiendo a una persona.

Es muy probable, casi seguro de hecho, que la consulta recibida formulada por el usuario no se corresponda con ninguna de las preguntas-tipo que generó el Ingeniero de Conocimiento cuando construyó las representaciones de los objetos del CCA. Por ello, será necesario modelar la consulta realizada extrayendo de ella aquellas palabras que corresponden a términos-índices presentes en el vocabulario del sistema y almacenados en la BD del CCA. Esto es, aquellas palabras que "tengan eco en la memoria" del sistema.

El procedimiento seguido es la simple comparación con los términos pertenecientes al vocabulario. De esta comparación se obtiene un array de términos-índices sin coeficientes de peso. A priori, no es posible saber como de importantes son los términos índices presentes en la consulta de usuario porque su valor puede variar dependiendo del ámbito al que se refiera.

3.3.2 Determinación del grado de relación de la consulta con los objetos

El procedimiento desarrollado parte de la consideración de presuponer que la expresión que el usuario ha enviado como consulta podría estar relacionada con un determinado objeto del CCA y evaluar la certeza de ello. Sería como si el método se preguntara: "Si estuviéramos hablando del objeto x, ¿cuánto sentido tendría lo que dice?"

Para dar respuesta a esta consideración el método toma los valores de peso de los términos-índices coincidentes entre la consulta y la representación del objeto

con el que se esté determinando el grado de relación y los asigna a la representación de la consulta. Si en la consulta aparecieran términos-índice que no existieran en la representación del objeto considerada se le asignaría un peso de valor 0. La entrada del sistema basado en FL desarrollado serán los pesos asociados a los términos índices que coincidan con las palabras extraídas de la consulta. El sistema devolverá un valor asociado a la certeza de la relación entre la consulta recibida y el objeto del CCA evaluado.

3.3.3 Algoritmo de recuperación de la información

En este punto es donde la estructura jerárquica del CCA propuesta en el apartado anterior cobra importancia. El nivel 0 del CCA contiene un único conjunto donde están englobados (contenidos) todos los objetos del CCA. Estos objetos se agrupan en subconjuntos de objetos similares. Los subconjuntos de objetos son representados por la unión de sus términos-índices los cuales tienen asociados, como ya vimos, unos pesos para este Nivel 1.

El sistema empezará evaluando la relación de la consulta con los subgrupos de Nivel 1 asignando a las entradas los pesos del subgrupo correspondiente como si de un objeto se tratara.

La entrada al sistema FL serán los pesos de los términos índices de cada subconjunto en este nivel presentes en la consulta recibida. El sistema devolverá un valor para cada subconjunto evaluado. Este valor es el grado de certeza asignado al subconjunto evaluado, de que la consulta esté relacionada con algún objeto de este subconjunto.

Si el grado de certeza asignado es inferior a un valor previamente fijado, denominado umbral, el subconjunto, y los objetos que contiene, es rechazado.

El propósito de la estructura jerárquica adoptada es identificar mediante una única evaluación grandes grupos de objetos del CCA que no guardan relación con la consulta recibida y rechazarlos aligerando la carga computacional del sistema y mejorando los tiempos de evaluación.

Para cada subconjunto cuyo grado de certeza supera el umbral fijado, se vuelve a realizar la evaluación usando la subdivisión y los coeficientes del nivel siguiente. En este caso, el Nivel 2. De nuevo se rechazan todos aquellos subconjuntos cuya certeza no supere el umbral fijado para este nivel y se vuelve a aplicar el procedimiento a los que lo superen. En este caso, estamos en el último nivel, el Nivel 3, cuyos subconjuntos son los propios objetos.

Se realiza una última evaluación y todos aquellos objetos cuya certeza supere el umbral fijado para este nivel serán señalados como respuesta a la consulta recibida. La Figura 7 muestra gráficamente el proceso descrito.

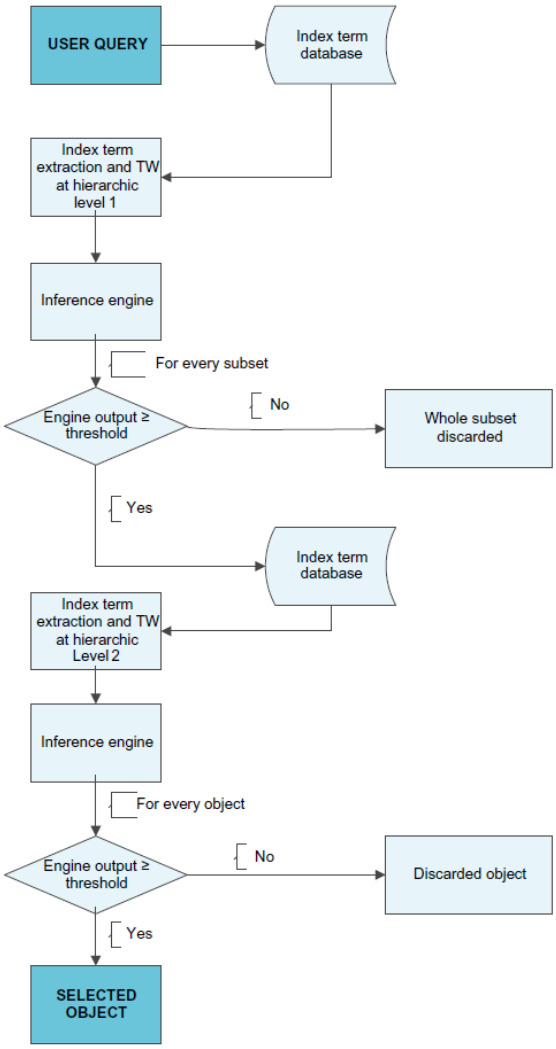


Figura 7: Proceso de recuperación de la información

4 Publicaciones

4 Publicaciones

En este capítulo se recopilan las publicaciones que sustentan la investigación realizada y se realiza un breve resumen de las aportaciones de las mismas

4.1 A Fuzzy Logic intelligent agent for Information Extraction: Introducing a new Fuzzy Logic-based term weighting scheme.

Breve descripción: En esta publicación se definen los parámetros de un nuevo método de recuperación de información al que se accede mediante Lenguaje Natural. También se define un método de asignación de pesos a los términos-índices. El núcleo de ambos métodos es un motor de lógica difusa. Como prueba de funcionamiento, se realizan pruebas sobre la recuperación de información aplicada a los contenidos del Portal web de la Universidad de Sevilla.

Autores: Jorge Ropero, Ariel Gómez, Alejandro Carrasco, y Carlos León..

DOI:10.1016/j.eswa.2011.10.009

Publicación: Expert Systems with Applications, Volume 39, Issue 4, March 2012, Pages 4567-4581.

Índice de calidad: La publicación está indexada en el JCR con índice de impacto de 1.854 (Q1) y el artículo ha recibido 6 citas.

Aportación personal: La aportación del doctorando en esta publicación consiste en la participación en el desarrollo del algoritmo de IR, la implementación del motor difuso para IR mediante el software Unfuzzy, determinación de reglas del motor difuso para IR y TW, determinación de los estimadores de calidad (métricas), diseño de pruebas de funcionamiento, análisis de resultados de dichas pruebas, optimización del algoritmo de IR, y colaboración en la redacción del artículo.

A Fuzzy Logic intelligent agent for Information Extraction: Introducing a new Fuzzy Logic-based term weighting scheme.

4.1.1 Introducción.

En este artículo se concretan los parámetros de un nuevo método para la recuperación de información (IR) en un Conjunto de Conocimiento Acumulado (CCA), con el fin de responder a las consultas formuladas en Lenguaje Natural (LN) de los usuarios. El núcleo del método es un motor de lógica difusa (Fuzzy logic Engine, FE), lo que añade flexibilidad en la exactitud de las consultas y tolerancia a fallos en la estructura de almacenamiento de la información. Como parte del proceso de indexación, el artículo también propone un método de asignación de pesos basado en lógica difusa.

Como prueba de funcionamiento, el método se aplicó para el caso de recuperación de información del portal web de la Universidad de Sevilla. Este portal contiene una enorme cantidad de información.

4.1.2 Objetivos.

El principal objetivo es especificar los parámetros de un método de Recuperación de Información basado en lógica difusa al que se acceda mediante Lenguaje Natural, capaz de obtener resultados satisfactorios incluso cuando la consulta incluya errores y/o vaguedades, y que mejore los resultados comparativamente con otros métodos existentes.

Un segundo objetivo es especificar un método de asignación de coeficientes de peso a los términos índices que representan a los objetos del Conjunto de Conocimiento Acumulado que, partiendo del método tf-idf y añadiendo dos nuevos parámetros, mejore la caracterización de los objetos de conocimiento y, por ende, los resultados de la posterior recuperación de la información.

Dentro de los objetivos marcados, también se encuentra la optimización del algoritmo de búsqueda de los contenidos y a resultas se propone una estructura de almacenamiento del CCA que consigue la optimización deseada.

4.1.3 Desarrollo y resultados.

Teniendo en cuenta que el propósito del sistema no es identificar sólo la mejor respuesta a la pregunta recibida sino devolver también aquellos objetos que se relacionen con la consulta, por si no hubiera sido correctamente formulada, parece

deseable agrupar los objetos mediante algún criterio de similitud. De esta forma el usuario podría recibir no solo el objeto identificado con mayor certeza sino también aquellos próximos a él.

En el caso de aplicación a un Portal web, cada página constituye en si misma un objeto aunque es frecuente que la información contenida en una página sea de tipo heterogéneo por lo que una misma página podría constituir varios objetos.

Tanto los objetos (sus representaciones) como la estructura jerárquica en la que se clasifican deben ser almacenados en la Base de Datos de su CCA según se definió en el apartado 3.3.1. La Figura 8 muestra la tabla de la Base de Datos que define la estructura jerárquica del CCA.

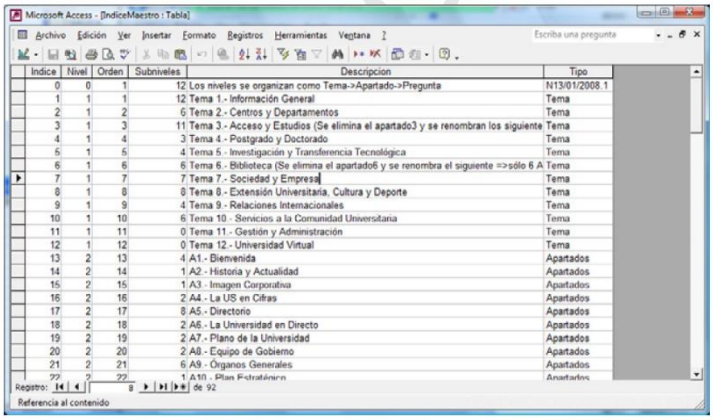


Figura 8: Tabla de definición de la estructura de la Base de Datos del CCA

4.1.4 Construcción del CCA.

En la construcción del Agente Inteligente, lo primero que hay que tener en mente es que las consultas de usuario se realizan en LN. Esta dificultad se convierte en una ventaja cuando la representación de los objetos se realiza mediante lo que denominamos las preguntas tipo y su agrupación se basa en la existencia de términos-índice comunes.

El primer paso es dividir todo el CCA en objetos y formular en LN una o varias preguntas tipo para cada objeto de forma que la respuesta a esas preguntas tipo sea el propio objeto a representar. En el caso de estudio, los objetos son los

contenidos del Portal web. En este punto, la experiencia del Ingeniero de Conocimiento que esté formulando las preguntas es importante, el conocimiento de las expresiones con las que los usuarios se refieran a los objetos que tiene que representar y la jerga de la temática en la que se encuentran aumentará el rendimiento del sistema. Esto no significa que las preguntas tipo sólo deban ser formuladas en términos técnicos o precisos, antes al contrario, deben ser incluidas todas aquellas formas en las que los usuarios, incluso los inexpertos, pudieran referirse al objeto.

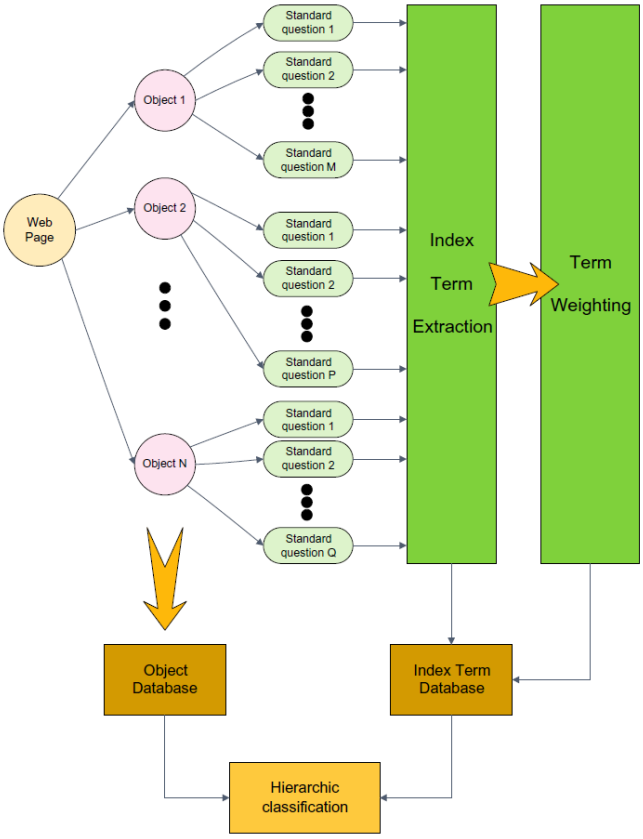


Figura 9: Proceso de construcción del CCA

Más adelante, una vez que el Sistema se encuentre en producción, la representación de los objetos puede ser redefinida ampliando o refinando las preguntas tipo asociadas como fruto de la experiencia de uso y análisis de las consultas formuladas por los usuarios.

El segundo paso consiste en seleccionar los términos índices que formarán parte de la representación del objeto. Estos términos índices serán las palabras o conjuntos de palabras contenidas en las preguntas tipo y que guarden una mayor relación con el objeto al que van a representar. Como la presencia de un determinado término índice no tiene por que ser igual de importante que la de otro, es necesario que exista un coeficiente asociado a cada término índice que indique de alguna manera el grado de importancia de su presencia en la asociación con el objeto. Dado que los objetos se agrupan formando conjuntos afines en diferentes niveles, es necesario que exista este coeficiente en cada nivel.

A dicho coeficiente se le suele denominar peso del término y al proceso cálculo de los pesos, o asignación de pesos (term weighting, TW). Para la asignación de estos pesos se consideran principalmente 2 métodos.

- Dejar que un experto en la materia asigne el valor de forma intuitiva basándose en su experiencia.
- Definir una serie de reglas y asignarlo de forma automática.

El primer método tiene la ventaja de ser muy simple pero es muy dependiente del Ingeniero de Conocimiento que lo esté haciendo y no se puede automatizar. Debido a la gran cantidad de contenido de un Portal web, proponemos realizar la asignación de pesos usando el segundo método.

El método clásico para la asignación de pesos más extendido es el conocido, y anteriormente mencionado, como tf-idf. En este artículo se propone una modificación de este método basado en el uso de la FL. Cada término índice tendrá asociado un peso en cada nivel de la estructura jerárquica de almacenamiento del CCA. El peso tendrá un valor comprendido entre 0 y 1 de forma que cuanto mayor es la importancia de la presencia de ese término para distinguir un objeto o grupo de objetos de otro en un nivel, mayor es el valor del peso asociado. Debido a que en cada nivel la agrupación de objetos es diferente, el peso asociado a un término en un determinado nivel puede ser diferente al que tenga en otro. Esto es consecuencia de que la presencia del término en un nivel puede ser determinante para la identificación y en otro irrelevante. Los términos índices y sus respectivos pesos en cada nivel deben ser almacenados en la base de datos de los contenidos del CCA.

4.1.5 Motor de inferencia.

El elemento del agente inteligente que determina el grado de certeza de que un grupo de términos índices pertenezca o no a un subconjunto de objetos del CCA es un motor difuso de inferencia. Este motor de inferencia tiene varias entradas cuyos valores corresponderán a los pesos de los términos índices considerados, y una salida cuyo valor corresponderá al grado de certeza de que dichos términos índices pertenezcan a objetos de ese subconjunto de un determinado nivel.

Para la definición del motor difuso de inferencia es necesario especificar:

- El número de entradas. Esto dependerá del número de términos índices presentes en la consulta, luego podría ser variable. También se podría forzar a que sea un número fijo y evaluar los términos con pesos mayores. De esta forma se pueden evitar consultas con pocos términos que sean muy vagas y devuelvan muchos objetos (o ninguno) como posible respuesta.
- Los conjuntos difusos de las entradas. Rango de valores de las entradas, número de conjuntos del Universo de Discurso, rango de cada conjunto, tipo y función de pertenencia.
- Los conjuntos difusos de la salida. Rango de valores de la salida, número de conjuntos del Universo de Discurso, rango de cada conjunto, tipo y función de pertenencia.
- Las reglas difusas. Reglas del tipo "SI … ENTONCES …. " como por ejemplo:
    - "SI todas las entradas son de valor Bajo, ENTONCES el valor de la salida es Bajo."
    - "SI una entrada es de valor Medio y las otras son de valor Bajo ENTONCES el valor de la salida es Medio-Bajo."
- El método utilizado como difusor de los valores reales.
- El método utilizado como concresor de los valores lingüísticos.

Es necesario fijar los valores de todos estos parámetros para obtener una optimización en la aplicación del método a los contenidos de un Portal web.

En este caso, se usan como entradas del sistema de FL los pesos del nivel superior de los términos índices iguales a las palabras extraídas de la consulta recibida.

4.1.6 Caso de aplicación.

Como caso de aplicación práctica que sirva para realizar la optimización de los valores de los parámetros del sistema, se toma el Portal Web de la Universidad de Sevilla.

Según el ranking web de universidades Webometrics, la Universidad de Sevilla se clasifica entre el 17% de las mejores a nivel mundial, ocupando la posición 200 entre las más de 11900 universidades consideradas. En la fecha de publicación del artículo, los datos consignaban que ocupaba el puesto 223 del ranking mundial entre más de 4000 universidades, con 50000 visitas diarias (Webometrics, 2009).

Sobre estos contenidos se realizarán las pruebas y en base a sus resultados se determinarán los valores de los parámetros del método general de IR basado en lógica difusa que mejor se adapten a este campo de aplicación.

Como la información contenida en el Portal utilizado como caso de aplicación es muy extensa, el CCA utilizado se restringe a los contenidos relacionados en su FAQ compuesta por 117 preguntas frecuentes de los usuarios. Usando estas preguntas frecuentes como contenido, se identifican 253 objetos. Estos objetos pueden corresponder a información contenida en alguna de las páginas web del Portal, o la página en si misma.

El número de preguntas tipo asociadas a cada objeto es variable, dependiendo de la cantidad de información contenida en cada página, su relevancia, y el número de sinónimos que tengan los términos de la pregunta tipo. Lógicamente, el conocimiento del administrador del sistema sobre el lenguaje de la materia es bastante importante. Cuanto mayor sea su conocimiento sobre el tema, mayor será la fiabilidad de las preguntas tipo propuestas, ya que serán más similares a las posibles consultas reales de los usuarios. En el caso de estudio, los 253 Objetos generan 2107 preguntas tipo.

También se mencionó previamente que los objetos del CCA debían ser agrupados por afinidad. En nuestro caso, los objetos se clasifican en 12 temas con un número variable de apartados por tema formando una estructura jerárquica de 3 niveles.

Una vez definidas las preguntas tipo, se extraen las palabras que mejor representan, señalan, o identifican a la pregunta tipo que las contiene. Estas palabras son los términos índice antes mencionados. Un término-índice también podría estar

compuesto por más de una palabra. Si este fuera el caso, se considerarían las palabras por separado y también el término compuesto. A continuación, se asocia un valor de peso entre 0.0 y 1.0 a ese término-índice de manera que cuanto más importante sea la presencia del término-índice para señalar la pregunta tipo, y por tanto al objeto, mayor debe ser su valor. Es de señalar que este valor no tiene por que ser el mismo en todos los niveles. Es posible que un término-índice sea muy determinante para señalar un grupo de objetos en un nivel determinado pero en el siguiente nivel no sirva para distinguir un objeto de otro. Por ejemplo en un CCA referido a menaje, "plástico" podría servir para distinguir el grupo de objetos de este material de todos los metálicos, o de madera, lo que haría que se le asignara un valor alto en este nivel de agrupamiento; pero para distinguir un tenedor de una cuchara o de un cuchillo todos de plástico, el término no tiene la menor relevancia por lo que se le asignaría un valor 0.0.

Un ejemplo de la metodología seguida se muestra en la Tabla 1

Tabla 1: Ejemplo del método de generación de términos-índices y pesos asociados

Step	**Example**
Step 1: Web page identified by standard question/s	– *Web page:* www.us.es/univirtual/internetwww.us.es/univirtual/internet
Step 2: Locate standard question/s in the hierarchic structure.	Topic 12: Virtual University Section 6: Virtual User Object 2
Step 3: Extract index terms	Index terms: 'services', 'virtual', 'user'
Step 4: Term weighting	See Section 6

El objetivo perseguido es recuperar el objeto u objetos relacionados con la consulta recibida. En el caso del ejemplo, la consulta realizada es "¿A qué servicios puedo acceder como usuario virtual de la Universidad de Sevilla?" y corresponde a una pregunta-tipo del objeto 12.6.2 (tema 12, apartado 6, objeto 2). Como respuesta el Agente identifica ese objeto y otros similares. En la Tabla 2 se muestran los resultados de la consulta.

Es evidente que identifica correctamente el objeto solicitado pero también ofrece al usuario otros objetos muy similares. Si recordamos que el usuario podría no ser experto en la búsqueda que realiza y preguntar algo aproximado, es posible que la respuesta exacta a lo que ha preguntado no satisfaga su necesidad pero que la respuesta deseada se encuentre entre alguna de las alternativas presentadas. Hay que tener en cuenta también que no es nada probable que el usuario formule una

consulta que coincida exactamente con alguna de las preguntas tipo del sistema por lo que este comportamiento flexible del sistema, motivado por el uso de la FL, responde exactamente a lo deseado.

Tabla 2: Resultados de la consulta de ejemplo

Position	Object	Certainty (%)	Associated standard question
1	12.6.2	74.13	Which services can I access as a virtual user at the University of Seville?
2	12.6.1	60.45	I would like to request for an account as a virtual user at the University of Seville
3	12.6.3	60.05	I do not remember my Virtual User password at the University of Seville
4	12.1.5	54.07	I would like to access the Economic Services at the Virtual Secretariat of the University of Seville
5	12.1.6	54.07	I would like to access the management services at the virtual secretariat of the University of Seville
6	10.4.9	48.96	What services does the service of computers and communications offer?
7	12.1.1	41.04	How can I access the virtual secretariat at the University of Seville?

4.1.7 Determinación de los parámetros del motor difuso.

Como ya se ha mencionado, el núcleo del Método desarrollado es un motor difuso de inferencia. En su implementación es necesario determinar parámetros tales como el número de entradas y salidas, los conjuntos difusos, y las reglas de inferencia. La forma de concretar los valores de estos parámetros es realizando pruebas de funcionamiento y concluyendo cuales son los valores que obtienen mejores resultados. La implementación del motor difuso sobre la que se realizan las pruebas está construida con Matlab y su toolbox de fuzzy logic, y el programa Unfuzzy. La primera prueba de funcionamiento es comprobar que el sistema realiza una identificación correcta de sus propias preguntas tipo obteniendo un índice de certeza no inferior al 0.7.

En sistemas de recuperación de información, los estimadores recall y precision ya descritos en el apartado 2.1.4 son los dos parámetros clásicamente utilizados para determinar la eficacia del comportamiento. El parámetro recall se relaciona con la cantidad de elementos correctamente identificados respecto al máximo número posible, mientras que el parámetro precision se relaciona inversamente con el número de elementos señalados como correctos pero que en realidad no lo son

(falsos positivos). Cuantos más elementos correctos se identifiquen, mayor es el valor del parámetro recall y mejor es el sistema. Así mismo, cuantos menos objetos erróneos sean identificados como relacionados con la consulta, mayor es el valor del parámetro precision y mejor es el comportamiento del sistema.

En el caso de nuestro sistema, dado que su objetivo es recuperar lo que identifica como relacionado y lo que identifica como similar, las expresiones exactas de los conceptos de recall y precision (Ruiz & Srinivasan, 1998) no son literalmente aplicables. Por ello, consideraremos los siguientes objetivos como estimadores de la calidad del sistema de IR:

- Como consultas de usuario usaremos las preguntas tipo que representan a los objetos del CCA del Sistema.
- Determinaremos si el sistema identifica la pregunta tipo con una certeza superior a un cierto valor prefijado (0,7). Este resultado está relacionado con el concepto de recall.
- Determinaremos si la pregunta tipo usada como consulta de usuario se encuentra entre las 3 identificadas con mayor certeza. Este parámetro se relaciona con el concepto de precision aunque no coincide exactamente con él.

De esta manera, establecemos 5 categorías para clasificar los resultados de las pruebas:

- Cat1. La pregunta correcta es la única que se recupera, o es la que se identifica con el grado de certeza más alto.
- Cat2. La pregunta correcta es la respuesta identificada con el segundo mayor grado de certeza.
- Cat3. La pregunta correcta es la respuesta identificada con el tercer mayor grado de certeza.
- Cat4. La pregunta correcta se identifica como respuesta, pero no entre las tres con mayor grado de certeza.
- Cat5. La pregunta correcta no se encuentra entre las identificadas como respuesta.

Como se ha mencionado, los pesos de los términos índices presentes en la consulta recibida serán los valores de entradas del motor difuso. Es evidente que el número de estos términos y sus pesos asociados y, por tanto, de entradas del motor

difuso, no serán constantes. Analizando las preguntas-tipo del sistema se puede observar que la mayoría tienen de 1 a 5 términos índice. Consideraremos que más de 5 términos pueden no ser relevantes en la IR. En el estudio se hicieron pruebas con 2 motores difusos, uno con 3 entradas, y otro con 5 entradas.

4.1.8 Resultado de las pruebas y ajuste de parámetros del motor difuso.

Las pruebas realizadas con la configuración de 3 entradas concluyen que el motor satura fácilmente, lo que constituye una desventaja atendiendo al parámetro precision. En la Tabla 3 se puede observar que con esta configuración se recupera el 90% de los objetos correctos pero menos de la mitad son identificados en 1ª opción.

En las pruebas realizadas con la configuración de 5 entradas se observa que, en general, los valores de certeza de la identificación obtenidos son menores que en el caso de la configuración de 3 entradas. El parámetro precision aumenta al 55% pero disminuye el valor del recall, lo que tampoco es deseable. Estos datos también pueden observarse en la Tabla 3.

Tabla 3: Resultados de las pruebas del motor difuso

Configuration	Cat1	Cat2	Cat3	Cat4	Cat5
Three-input fuzzy engine results	45%	24%	9%	12%	10%
Five-input fuzzy engine results.	55%	12%	3%	1%	29%
Five-input fuzzy engine results with variable output thresholds.	70%	14%	3%	1%	12%
Five-input fuzzy engine results with variable output thresholds and variable input number fuzzy engine.	77%	16%	4%	1%	2%

Analizando los casos fallidos también se encuentra para ambas configuraciones que muchos casos en los que el sistema no devuelve respuesta es debido a que, en alguna etapa del proceso, la certeza de los objetos correctos estaba por debajo del umbral mínimo de aceptación definido para ese nivel. Como solución a este problema, podría disminuirse el valor de los umbrales para que esas respuestas los superaran y no fueran descartadas. No obstante, hacer esto también conllevaría que en las preguntas correctamente identificadas se dieran por válidos objetos actualmente descartados lo que también empeoraría el parámetro precision, cosa no deseable.

La conclusión extraída de los resultados observados en las pruebas realizadas es que configurar el motor difuso con un nº bajo de entradas o bajar el valor de los umbrales afecta negativamente al parámetro precision mientras que si se configura un valor alto de entradas o se eleva el valor de los umbrales, el parámetro que se ve negativamente afectado es el recall.

Ante estos resultados, la primera acción correctiva adoptada es introducir en el algoritmo de recuperación de la información una bajada automática del valor de los umbrales en el caso de que ningún objeto supere el valor exigido. Los resultados considerados en la Tabla 3 muestran una notable mejora. Considerando los 3 objetos recuperados con mayor certeza, el objeto correcto aparece el 88% de los casos y es la 1ª opción el 70% de las veces.

De los resultados anteriores también se desprende que unas veces es mejor usar la configuración de 3 entradas y otras la de 5 entradas. La siguiente acción correctiva introducida es utilizar un motor con 3 entradas cuando el nº de términos índices presentes en la consulta recibida sea menor o igual a 3, y un motor de 5 entradas cuando se identifican más de 3 términos índices. Repitiendo las pruebas incluyendo esta segunda acción correctora y volviendo a considerar como objetivo que el objeto correcto aparezca entre los 3 identificados con mayor probabilidad, el objeto correcto se recupera el 97% de los casos, y es señalado con mayor índice de certeza el 77%, como se muestra en la Tabla 3.

Consideramos en adelante como configuración final la que incorpora los umbrales de valor variable de forma automática, y la configuración de un motor difuso de 3 ó 5 entradas dependiendo del nº de términos índices identificados en la consulta de usuario.

En las pruebas realizadas se ha considerado 0.5 como valor umbral en todos los niveles.

4.1.9 Conjuntos difusos.

Como ya se ha mencionado anteriormente, el valor del peso asignado a cada término índice se encuentra en el rango de 0.0 a 1.0. Por lo tanto, ese será también el rango del valor de las entradas del motor difuso y constituirá el universo de discurso de la variable correspondiente.

Por motivos de minimización del nº de reglas a definir, ya que si aumentamos el nº de conjuntos difusos el nº de reglas aumenta exponencialmente, se considera inicialmente una división del Universo de Discurso en 3 conjuntos difusos lo que arroja para el motor de 5 entradas la necesidad de 3^5=243 reglas. En el caso de

dividir el universo de discurso de las entradas en 4 conjuntos difusos en vez de en 3, el nº de reglas necesarias para el motor difuso de 5 entradas pasaría a ser de 4^5=1024. Si posteriormente las pruebas indicaran un funcionamiento con pobres resultados, se pasaría a aumentar el nº de conjuntos difusos para aumentar la granularidad de la respuesta.

Por todo ello, este Universo de discurso se considera dividido en tres conjuntos difusos cuya representación lingüística será BAJO, MEDIO, y ALTO. Inicialmente se parte de considerarlos de tipo triangular, también por simplicidad, y más adelante se realizarán pruebas comparativas para determinar el tipo de frontera que mejor resultados ofrezca.

Los rangos de valores asignados a cada conjunto difuso de las entradas son los siguientes:

BAJO: de 0.0 a 4.0 con vértice en 0.0

MEDIO: de 0.2 a 0.8 con vértice en 0.5

ALTO: de 0.6 a 1.0 con vértice en 1.0

La Figura 10 muestra el aspecto de estos conjuntos difusos

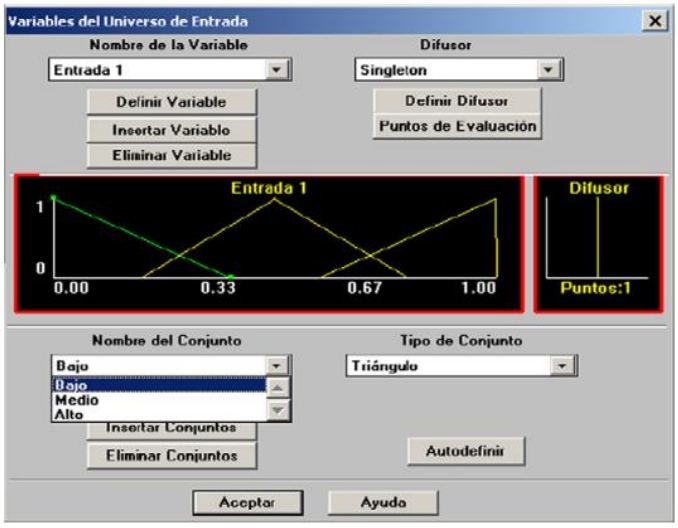


Figura 10: Conjuntos difusos de las entradas

La salida, que corresponde al valor de certeza de relación del objeto con la consulta recibida, también tendrá un rango de valores entre 0.0 y 1.0. Inicialmente su Universo de Discurso también se clasifica en tres conjuntos difusos.

No obstante, en el caso de la salida, las pruebas iniciales rápidamente indican la necesidad de una mayor resolución en la clasificación de sus valores por lo que finalmente se opta por dividirlo en 4 conjuntos difusos. La representación lingüística será BAJO, MEDIO-BAJO, MEDIO-ALTO, y ALTO

Los rangos de valores asignados a cada conjunto difuso de la salida son los siguientes:

BAJO: de 0.0 a 4.0 con vértice en 0.0

MEDIO-BAJO: de 0.1 a 0.7 con vértice en 0.4

MEDIO-ALTO: de 0.3 a 0.9 con vértice en 0.6

ALTO: de 0.6 a 1.0 con vértice en 1.0

4.1.10 Reglas difusas.

En un motor difuso, una vez conocidos el nº de entradas y salidas, es necesario definir las reglas difusas que determinarán los valores de las salidas según los de las entradas. Como se comentó anteriormente, el Sistema utilizará un nº de entradas variable (3 ó 5) según el número de términos índices identificados en la consulta.

Las reglas necesarias corresponden al nº de conjuntos difusos del universo de discurso de las entradas elevado al nº de entradas.

Nº Reglas = Nº Conjuntos $^{\text{NºEntradas}}$

En el caso del motor de 3 entradas es necesario definir $3^3 = 27$ reglas. Para el caso del motor de 5 entradas serán $3^5 = 243$ reglas.

En la práctica esto lleva a la implementación de dos motores de inferencia diferentes, uno con 3 entradas y otro con 5.

A modo de ejemplo, las reglas difusas definidas para el motor difuso de 3 entradas pueden observarse en la Tabla 4. La concreción de estas reglas da lugar a las 27 combinaciones necesarias.

Tabla 4: Reglas para el motor de 3 entradas

Rule number	Rule definition	Output
R1	IF one or more inputs = HIGH	HIGH
R2	IF three inputs = MEDIUM	HIGH
R3	IF two inputs = MEDIUM and one input = LOW	MEDIUM–HIGH
R4	IF one input = MEDIUM and two inputs = LOW	MEDIUM–LOW
R5	IF all inputs = LOW	LOW

4.1.11 Comparación de ambos métodos de asignación de pesos.

Para comparar el rendimiento del método desarrollado, se realiza una prueba de rendimiento y se compara con el obtenido utilizando el método tf-idf.

De entre las diversas modificaciones del estimador tf-idf, para nuestro estudio elegimos la propuesta por Liu et al., 2001 para calcular el peso W_{ik} asignado a un término t_i en un subconjunto n_k.

$$W_{ik} = \frac{tf_{ik} \times \log\left(N / n_k + 0.01\right)}{\sqrt{\sum_{k=1}^{m} tf_{ik} \times \left(N / n_k + 0.01\right)^2}}$$

Donde tf_{ik} es la frecuencia de aparición del i-ésimo término en el k-ésimo conjunto (tema/apartado/objeto); n_k es el número de subconjuntos que contienen el término t_i, y N es el número total de subconjuntos (en ese nivel).

Hay que tener en cuenta que un término t_{ik} también puede estar contenido en otros conjuntos del Nivel. Como ejemplo, consideraremos el término “virtual”.

En el nivel de tema:

- Aparece 8 veces en el tema 12 (k = 12, tf_{ik} = 8)
- Aparece 2 veces en otros Temas (n_k = 3)
- Hay 12 Temas (subconjuntos) en este nivel (N = 12)
- Sustituyendo W_{ik} = 0.20 (para normalizar, es necesario conocer los otros tf_{ik}, y n_k para los otros Temas en los que aparece).

En el nivel de apartado:

- Aparece 3 veces en el apartado 6 del tema 12 (k = 6, tf_{ik} = 3)
- Aparece 5 veces en otros apartados del tema 12 (n_k = 6)

- Hay 6 apartados (subconjuntos) en el tema 12 (N = 6)
- Sustituyendo W_{ik} = 0.17 (para normalizar, es necesario conocer los otros tf_{ik}, y n_k para los otros apartados en los que aparece).

En el nivel de objeto:

- Aparece 1 vez en el objeto 2 del apartado 6 del tema 12 (k = 2, tf_{ik} = 1). Lógicamente, un término sólo aparece una vez en cada objeto.
- Aparece 2 veces en otros objetos del apartado 6 del Tema12 (n_k = 3)
- Hay 3 objetos (subconjuntos) en el apartado 6 en el tema 12 (N = 3)
- Sustituyendo W_{ik} = 0.01 (para normalizar, es necesario conocer los otros tf_{ik}, y n_k para los otros apartados en los que aparece). Obsérvese que este término es irrelevante para distinguir el objeto de los demás, lo cual es lógico ya que, según se desprende de los datos, aparece en todos ellos.

Por ello, se puede concluir que el término "virtual" será relevante para determinar que el objeto pertenece al tema 12 y al apartado 6 pero no en la determinación del objeto individual. La determinación a nivel de objeto deberá ser realizada en base a los otros términos presentes en la consulta.

Para determinar el peso correspondiente mediante el método FL, es necesario contestar a las cuatro preguntas mencionadas en el apartado 3.1 y que se formulan a continuación.

- P1.- ¿En cuántos subconjuntos diferentes al propio aparece el término evaluado? Esta pregunta está relacionada con el concepto idf.
- P2.- ¿Cuántas veces aparece el término evaluado en su subconjunto? Esta pregunta está relacionada con el concepto tf.
- P3.- El término evaluado, ¿identifica inequívocamente por si mismo al objeto?
- P4.- ¿A cuántas palabras está unido el término evaluado para formar un término compuesto?

Los valores que se usarán como entradas del sistema basado en FL que devolverá el peso asignado están relacionados con las respuestas a estas preguntas. Es de recordar que el rango de valores de las entradas es 0.0 a 1.0 por lo que los

valores obtenidos como respuesta de las preguntas deben ser normalizados y adaptados al rango. La Figura 11 muestra el esquema.

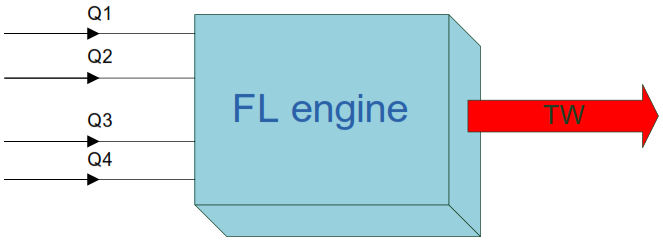


Figura 11: Asignación de pesos. Método basado en FL

Para la pregunta P1, si el término considerado aparece muchas veces, el valor asociado debe ser 0.0 y si no aparece ninguna debe ser 1.0. Para establecer el valor numérico correspondiente al concepto de "muchas veces", y la escala intermedia, se considera el número de subconjuntos de un determinado nivel en los que aparece cada uno de los términos-índice del vocabulario del CCA y se ordenan de mayor a menor número de apariciones. Por ejemplo, para el nivel de tema, se considera el número de apariciones de cada término del CCA en cada tema y se clasifican en orden descendente según su frecuencia de aparición:

1.- Servicio: 31 apariciones

2.- Servicios: 18 apariciones

3.- Biblioteca: 16 apariciones

4.- Investigación: 15 apariciones

5.- Dirección: 14 apariciones

6.- Estudiante: 14 apariciones

7.- Correo: 13 apariciones

8.- Acceso: 13 apariciones

9.- Electrónico: 12 apariciones

10.-Ordenador: 12 apariciones

11.- Recursos: 12 apariciones

12.- Centro: 10 apariciones

13.- Educación: 10 apariciones

14.- Registro: 10 apariciones

15.- Programa: 10 apariciones

Como ya se enunció en el apartado 3.1.1, el método desarrollado propone que la frecuencia de aparición del 1% de los términos-índice del vocabulario que aparecen con mayor asiduidad debe señalar el valor a partir del cual el coeficiente debe valer 0.0. Teniendo en cuenta que el vocabulario del CCA considerado lo forman 1114 términos-índices, la frecuencia de aparición del término-índice que ocupe el undécimo lugar en la ordenación anterior establecerá el valor frontera a partir del cual se asignará un valor de 0.0 al coeficiente. En nuestro caso corresponde a 12 apariciones. La Tabla 5 muestra el coeficiente asignado a cada valor de aparición.

Tabla 5: Valores asignados a la P1 según el nº de apariciones. Nivel de tema

Times appearing	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1 0**	**1 1**	**1 2**	**≥13**
Value	1	0.9	0.8	0.7	0.64	0.59	0.53	0.47	0.41	0.36	0.3	0.2	0.1	0

Si el término índice aparece más de 12 veces en otros subgrupos, el valor asignado debe ser 0.0. Si aparece de 0 a 3 veces (1/3 aproximadamente del rango) se considerará que pertenece al grupo ALTO asignándosele valores entre 0.0 y 1.0. Operando de forma análoga para el conjunto BAJO, y dejando el resto de valores para el MEDIO, se obtienen los valores de la Tabla 5.

Este proceso debe ser calculado para cada nivel ya que la división en subconjuntos es diferente, lo que dará lugar a ordenaciones distintas. Para el nivel de apartado, hay que considerar el nº de veces que los términos índices de ese apartado aparecen dentro de su tema, y hacerlo para todos los temas. La lista de los términos índices más usados en cada tema se muestra a continuación.

1.- Servicio en tema 10: 16 apariciones

2.- Direccion en tema 1: 10 apariciones

3.- Biblioteca en tema 6: 10 apariciones

4.- Registro en tema 3: 10 apariciones

5.- Correo en tema 1: 9 apariciones

6.- Electronico en tema 1: 9 apariciones

7.- Virtual en tema 12: 8 apariciones

8.- Ordenador en tema 10: 7 apariciones

9.- Servicios en tema 1: 7 apariciones

10.- Educación en tema 1: 5 apariciones

11.- Recursos en tema 12: 5 apariciones

De igual forma que se procedió en el nivel de tema, usaremos el 1% de las apariciones de mayor frecuencia para determinar el valor frontera. En este caso el valor corresponde a 5. Si un término aparece más de 5 veces en otros conjuntos, el valor asignado será 0. Si un término sólo aparece en su subconjunto (apartado) el valor asignado será 1. La tabla que asigna los valores se construye de igual forma que en el caso del nivel anterior (tema). Los valores calculados se muestran en la Tabla 6.

Tabla 6: Valores asignados a la P1 según el nº de apariciones. Nivel de apartado

Times appearing	**0**	**1**	**2**	**3**	**4**	**5**	**≥6**
Value	1	0.7	0.6	0.5	0.4	0.3	0

Para el nivel de objeto se procede de forma análoga y se obtienen los resultados mostrados en la Tabla 7.

Tabla 7: Valores asignados a la P1 según el nº de apariciones. Nivel de objeto

Times appearing	**0**	**1**	**2**	**≥3**
Value	1	0.7	0.3	0

Para averiguar el valor asociado a la pregunta P2.- ¿Cuántas veces aparece el término evaluado en su subconjunto? El razonamiento es análogo debiendo considerar de nuevo el 1% de los términos ordenados para establecer el valor frontera aunque con las siguientes particularidades:

- Cuanto más veces aparece el término en el subconjunto considerado, mayor es el valor que hay que asignarle.
- Este término no tiene sentido en el nivel de objeto dado que todos los conjuntos son unitarios y, por tanto, el término sólo aparece una vez.

En este caso, la frecuencia de aparición del término que establece el valor frontera es la misma para el nivel de tema y para el de apartado. Este resultado es completamente casual. La Tabla 8 contiene los valores propuestos.

Tabla 8: Valores asignados a la P2 según el nº de apariciones. Nivel de tema y apartado

Times appearing	**0**	**1**	**2**	**3**	**4**	**5**	**≥6**
Value	0	0.3	0.4	0.5	0.6	0.7	1

Para el caso de la pregunta P3.- El término evaluado, ¿identifica inequívocamente por si mismo al objeto? La respuesta es completamente subjetiva y se proponen tres posibles respuestas: "Sí", "Algo", y "No". La Tabla 9 muestra los valores asociados.

Tabla 9: Valores asignados a la P3

Answer to Q3: Does a term define undoubtedly a standard question?	**Yes**	**Rather**	**No**
Value	1	0.5	0

Por último, en el caso de la pregunta P4.- ¿A cuántas palabras está unido el término evaluado para formar un término compuesto? Se consideran cuatro posibles respuestas atendiendo al número de términos a los que esté unido: "A ninguna", A otra", "A dos", "A más de dos". Los valores asociados a cada una de ellas se muestran en la Tabla 10. De nuevo, los valores 0.7 y 0.3 son consecuencia de considerar la frontera entre conjuntos difusos.

Tabla 10: Valores asignados a la P4

Number of index terms tied to another index term	**0**	**1**	**2**	**≥3**
Value	1	0.7	0.3	0

El único aspecto que no se ha definido es el valor que toma un coeficiente cuando el término aparece varias veces en un mismo subconjunto (tema o apartado) con valores diferentes. Por ejemplo, puede ocurrir que la respuesta a la pregunta P3 sea "Algo" en un caso y "No" en otro. En este caso, el valor asignado se calcularía mediante una media ponderada de los valores individuales.

A continuación se muestra un ejemplo de todo el proceso. Usaremos como ejemplo el objeto 2 del apartado 6 del tema 12 (objeto 12.6.2). Este objeto está definido por la pregunta tipo: "¿A que servicios puedo acceder como usuario virtual de la Universidad de Sevilla?"

Si aplicamos el proceso de asignación de peso al término-índice "virtual":

A nivel de tema el término-índice "virtual" aparece 2 veces en otros temas por lo que, según la Tabla 5, el valor asociado a P1 es 0.80; y aparece 8 veces en el

tema 12 (su tema) por lo que, según la Tabla 8, el valor asociado a P2 es 1. La respuesta a la P3 es "Algo" en 5 de las 8 apariciones y "No" en las otras 3 por lo que, según la Tabla 9 y considerando la media de los valores individuales, el valor asignado a la P3 es de (5*0.5 + 3*0)/8=0.375. El término "virtual" está ligado a otro término en 7 de las apariciones y a otros 2 en una ocasión por lo que la media de términos ligados es de 1.14. Haciendo una regresión lineal o una media de los valores individuales de la Tabla 10, el valor obtenido para el parámetro P4 es de 0.65.

Con estos valores en las entradas, el peso que asigna el motor difuso a este término es de 0.53.

Para el nivel de apartado, "virtual" aparece 5 veces en otros apartados del tema 12 por lo que, según la Tabla 6, el valor asociado a P1 es de 0.30; y aparece 3 veces en el tema 12 por lo que, según la Tabla 8, el valor asociado para P2 es de 0.4. La respuesta a la pregunta P3 es "Algo" en todo los casos por lo que el valor asociado es 0.5. El término "virtual" está asociado al término "usuario" por lo que el valor de P4 es 0.7.

Con estos valores en las entradas, el peso que asigna el motor difuso a este término es de 0.45.

Para el nivel de objeto, "virtual" aparece 2 veces en otros objetos del apartado 6 del tema 12 por lo que, según la Tabla 7, el valor asociado a P1 es 0.3. La respuesta a la pregunta P3 es "Algo" por lo que el valor asociado será de 0.5. El término "virtual" está asociado al término "usuario" por lo que el valor establecido para P4 es de 0.7.

Con estos valores en las entradas, el peso que asigna el motor difuso a este término es de 0.52.

Aplicando los estimadores clásicos del método tf-idf se puede observar, como era de esperar, que hay una diferencia de valores en el peso asignado mediante el método tf-idf y el método basado en FL. Precisamente eso es lo que se pretende para que luego el método de recuperación devuelva, no sólo el objeto que mejor corresponda, sino también aquellos relacionados.

Para realizar la comparación de eficiencia de ambos métodos se categorizan los resultados obtenidos por cada uno de ellos según la clasificación en 5 categorías especificada en el apartado 0.

El resultado idóneo es que la respuesta del método se clasifique en la categoría Cat1 aunque los resultados clasificados en Cat2 o Cat3 también son razonablemente buenos. Los resultados obtenidos de usar todas las preguntas tipo como consultas de usuario se recogen en la Tabla 11.

Tabla 11: Comparación de resultados aplicando método tf-idf vs FL

Method	Cat1	Cat2	Cat3	Cat4	Cat5	Total
TF-IDF	466	223	53	79	93	914
	(50,98%)	(24,40%)	(5,80%)	(8,64%)	(10,18)	(100%)
FL	710	108	27	28	41	914
	(77,68%)	(11,82%)	(2,95%)	(3,06%)	(4,49%)	(100%)

Aunque los resultados obtenidos con el método tf-idf son razonablemente buenos con un 81.18% de los Objetos identificados en las 3 primeras categorías, y la mayoría en Cat1, el método basado en FL propuesto es claramente mejor con un 92.45% de objetos correctos en las 3 primeras categorías y más del 75% identificados como la opción más probable.

4.1.12 Conclusiones.

En este artículo se elige el ámbito de un Portal Web como caso de uso de un nuevo método de IR basado en FL.

Se realizan pruebas y del análisis de resultados se concretan los parámetros de sistema consiguiendo resultados superiores a los que se obtienen aplicando el método clásico de asignación de pesos basada en el método tf-idf.

La principal conclusión obtenida es que los Portales Web, en los que suele haber mucha información heterogénea y en ocasiones confusa, son un ámbito de aplicación donde el método desarrollado basado en FL ofrece muy buenos resultados.

4.2 SABIO: Soft Agent for Extended Information Retrieval.

Breve descripción: En esta publicación se describe la implementación de un Agente Inteligente para recuperación de información utilizando los métodos desarrollados y detallados en publicaciones anteriores. Se programa el motor difuso, sus reglas, y los algoritmos de recuperación de información, utilizando el entorno Borland C++ Builder. Se implementa una base de datos para almacenar el conocimiento del Agente y sus parámetros mediante Access. Se diseñan pruebas de funcionamiento y se optimiza el algoritmo de recuperación mediante análisis de los resultados.

Autores: Ariel Gómez, Jorge Ropero, Alejandro Carrasco, Carlos León, y Joaquín Luque.

DOI: 10.1080/08839514.2013.774204

Publicación: Applied Artificial Intelligence, Volume 27, Issue 4, 1 April 2013, Pages 249-277. ISSN:0883-9514; EISSN:1087-6545.

Índice de calidad: La publicación está indexada en el JCR con índice de impacto de 0.402 (Q4) y el artículo ha recibido 1 cita.

Aportación personal: La aportación del doctorando en esta publicación consiste en el desarrollo de varios FE en Unfuzzy, implementación del FE genérico en C, la participación en la definición e implementación de las reglas para el motor difuso de IR y TW, la implementación del Agente Inteligente para IR, optimización del algoritmo de IR programado, implementación del Agente asignador de coeficientes de peso, diseño de la estructura e implementación de una Base de Datos para almacenamiento del CCA del sistema, diseño de pruebas del Agente de IR y del Agente asignador de pesos, análisis de resultados de las pruebas, y redacción del artículo.

SABIO: Soft Agent for Extended Information Retrieval.

En esta publicación se describe la implementación de un Agente Inteligente para la recuperación de Información que utiliza el método de asignación de pesos y de recuperación de información desarrollado en esta tesis.

El método de asignación de pesos y el de recuperación de información, ambos basados en lógica difusa, se ha desarrollado y comprobado utilizando el software Matlab y su toolbox de lógica difusa, y el programa de libre distribución Unfuzzy para la generación de los motores difusos y los juegos de reglas utilizados.

Una vez llegados a establecer los métodos y configuraciones desarrollados en apartados anteriores de esta tesis, se da el paso de implementar estos conceptos mediante la programación de una aplicación informática. En esta aplicación, el motor de inferencia, las reglas, y los algoritmos de recuperación de información se programan utilizando Borland C++ Builder. La base de datos contenedora del CCA del sistema se implementa mediante Microsoft Access.

Además, para la comprobación del correcto funcionamiento del método de asignación de pesos basado en lógica difusa desarrollado en esta tesis se construye una maqueta del asignador y se comparan los valores propuestos por esta aplicación con los inicialmente asignados por el Ingeniero de Conocimiento y que mejoraron los resultados de determinación comparativamente con el método tf-idf, como se justifica en el apartado anterior.

La maqueta realizada mediante Borland C++ Builder también se utiliza para realizar las pruebas de funcionamiento del Agente implementado. La última versión de esta maqueta es un sistema cliente-servidor que permite realizar la extracción de información desde una aplicación cliente ubicada en un PC remoto que conecta vía http con la aplicación servidor que constituye el Agente implementado.

4.2.1 Objetivos.

El objetivo de este trabajo es comprobar la viabilidad de un Agente Inteligente basado en los métodos de TW y recuperación de información desarrollados en capítulos previos de esta tesis implementado mediante una aplicación programada en Borland C++ Builder y con el que el usuario interactúe en Lenguaje Natural.

Este objetivo global se puede descomponer en los siguientes objetivos parciales:

- Implementación de dos motores difusos con configuración de parámetros variable. Serán parametrizables el concresor, el difusor, el tipo de cada conjunto difuso de entrada, y el tipo de cada conjunto difuso de salida.
- Definición de las reglas de los motores difusos. Se especifican un juego de reglas para cada motor desarrollado tanto para cuando se utilizan en el cálculo de los valores de los pesos de los términos índices como para cuando se utilizan para establecer el grado de certeza de la relación de la consulta recibida con los subconjuntos del CCA en un nivel determinado.
- Implementación en C del algoritmo de recuperación de información desarrollado en los capítulos anteriores de esta tesis permitiendo parametrizar, la elección entre el motor difuso de 3 entradas, el de 5, o la configuración dependiente del nº de términos índices identificados en la consulta; y el valor del umbral de aceptación de cada nivel.
- Implementación en Microsoft Access de una Base de Datos que almacene el CCA del Agente. Esta BD deberá contener el vocabulario del Agente categorizado por tipología de términos (términos índices, palabras de interés gramatical, términos compuestos, palabras soeces, etc), los coeficientes de peso asignados a cada término índice para cada nivel, las preguntas-tipo asociadas a cada objeto, las respuestas asociadas a cada objeto a entregar al usuario, otros datos de configuración general.
- Acceso de usuario mediante Lenguaje Natural.
- Estructura cliente-servidor para comprobar el acceso remoto a la aplicación mediante un terminal no inteligente.

4.2.2 Motor Difuso.

Dado que los métodos desarrollados en esta tesis se basan en razonamientos de lógica difusa, el corazón de la aplicación para implementar el Agente Inteligente será un motor de lógica difusa que dé soporte a los razonamientos propuestos.

Para ello, se implementa en la aplicación un motor difuso genérico programado en C, basado en el motor difuso desarrollado con Unfuzzy con el cual

se realizó gran parte del desarrollo y pruebas de los métodos desarrollados en esta tesis.

Este motor se programa de forma que los parámetros de difusor, concresor, tipo de conjuntos difusos de entrada, y tipo de conjuntos difusos de salida, puedan ser definidos con posterioridad para que sea posible modificar la configuración y buscar la que ofrezca mejores resultados.

No obstante, en la implementación realizada no es posible parametrizar el número de entradas ni el número de conjuntos en los que se divide su universo de discurso, por lo que será necesaria una instancia diferente del motor difuso si se necesita un número de entradas distinto. El origen de esta falta de flexibilidad reside en que la implementación de las reglas difusas genera un número de casos que dependen del número de entradas y del número de conjuntos de su universo de discurso.

4.2.2.1 Reglas del motor difuso para el cálculo de los pesos.

Como ya se mencionó en apartados anteriores, el método de asignación de pesos desarrollado en esta tesis introduce dos parámetros de tipo no estadístico cuyo valor necesita ser cuantificado por una persona. La inferencia del valor final del coeficiente de peso a asociar al término índice no tiene una relación numérica fácil de obtener mediante la aplicación de fórmulas matemáticas a los valores de los parámetros definidos por el método. Por ello, su valor se obtendrá de la aplicación de la lógica difusa. Se utiliza un motor difuso cuyas entradas serán los valores de los 4 parámetros especificados, y su salida el valor asignado al coeficiente de peso correspondiente.

Las entradas del motor difuso corresponden a los valores asociados a las 4 preguntas definidas. Los posibles valores de entrada son: BAJO, MEDIO, o ALTO. La salida corresponderá al valor de peso asociado. Los posibles valores de salida serán BAJO, MEDIO-BAJO, MEDIO-ALTO, o ALTO. La Tabla 12 recoge las reglas definidas y los valores asociados.

En la Figura 12 se muestra el aspecto del primer prototipo creado para comprobar la repercusión de las reglas en los valores de los coeficientes de peso.

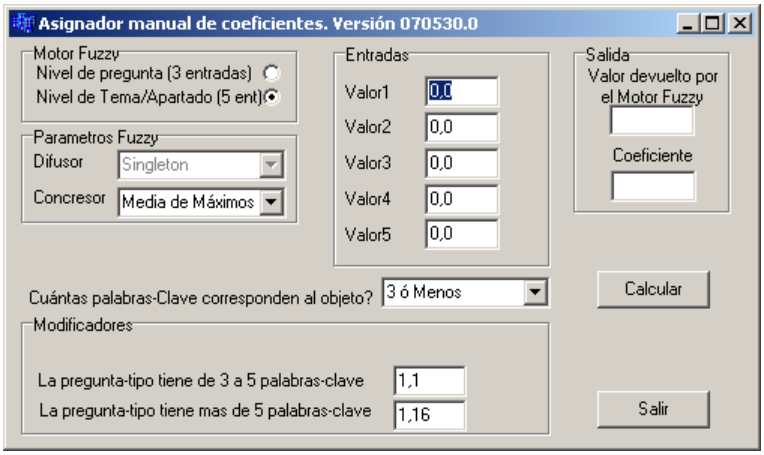


Figura 12: Prototipo de asignador de pesos

Tabla 12: Reglas para el cálculo de pesos

Regla Nº	**Definición de la Regla**	**Salida**
R1	Si P2 = ALTO, y P3 ≠ BAJO	Al menos MEDIO-ALTO
R2	SI P2 = MEDIO, y P3 = ALTO	Al menos MEDIO-ALTO
R3	SI P2 = BAJO, y P3 = BAJO	Depende de las otras preguntas
R4	SI P2 = ALTO, y P3 = ALTO	Depende de las otras preguntas
R5	Si P1 = ALTO	Al menos MEDIO-ALTO
R6	Si P4 = BAJO	Baja un nivel
R7	Si P4 = MEDIO	Si la salida era MEDIO-BAJO, pasa a BAJO
R8	Si (R1 y R2) o (R1 y R5) o (R2 y R5)	ALTO
R9	Cualquier otro caso	MEDIO-BAJO

Como ya se explicó en apartados anteriores de esta tesis, el valor asignado al parámetro relativo a cuánto identifica el término índice al que se le está calculando el coeficiente por sí mismo al objeto debe ser definido por un Ingeniero de Conocimiento y es completamente subjetivo, por lo que las asignaciones de este valor pueden variar apreciablemente dependiendo de quien realice la valoración. Para minimizar la variación introducida por este parámetro, se plantea que el valor numérico asociado al parámetro en cuestión no lo fije el IC sino que éste responda a la siguiente pregunta: ¿El término considerado identifica inequívocamente al objeto cuando aparece solo? Las posibles respuestas entre las que puede optar el IC son: Sí, Algo, o No. El valor asociado a este parámetro será el mostrado en la tabla siguiente.

Tabla 13: Valor del parámetro P3

¿El término considerado identifica inequívocamente al objeto cuando aparece solo?	**Sí**	**Algo**	**No**
Valor del 1er parámetro	1	0.5	0

El parámetro 4 correspondiente al número de términos índices asociados al que se está evaluando, también debe ser introducido por el IC. En esta ocasión se vuelve a sustituir la tarea de proponer un valor que dependerá de su criterio personal y que podría sufrir fuertes diferencias por la de indicar el dato objetivo del número de términos índices ligados al que se está evaluando. La Tabla 14 muestra los valores asociados a la casuística considerada.

Tabla 14:Valor del parámetro P4

Numero de términos ligados al evaluado	**0**	**1**	**2**	**≥3**
Valor del 4º Parámetro	1.00	0.70	0.30	0.00

4.2.2.2 Optimización del motor difuso.

Para determinar la configuración de los parámetros del motor difuso que consigue unos resultados óptimos, se planifican 6 pruebas realizadas con distintas combinaciones de los siguientes parámetros: Difusor, concresor, tipo de conjuntos difusos de entrada, tipo de conjuntos difusos de salida. La Tabla 15 muestra los valores de dichos parámetros que se adoptarán en cada auto-test.

Tabla 15: Parámetros de los auto-test

Test nº	**Difusor**	**Concresor**	**Universo de Entrada**	**Universo de Salida**
1	Singleton	*CdG*	Recto	Recto
2	Triangulo	CdG	Recto	Recto
3	Singleton	MdM	Recto	Recto
4	Singleton	CdG	Curvo	Curvo
5	Triangulo	CdG	Curvo	Curvo
6	Singleton	MdM	Curvo	Curvo

Los resultados obtenidos se muestran en la Tabla 16

Tabla 16: Resultados de los auto-test

Test nº	Cat1	Cat2	Cat3	Cat4	Cat5
1	77.44	15.79	4.51	0.75	1.51
2	69.17	18.05	3.76	5.26	3.67
3	68.42	15.04	6.77	7.16	2.26
4	75.94	15.79	4.51	1.50	1.50
5	84.21	8.21	1.50	2.26	3.76
6	65.41	18.78	6.02	8.27	1.50

De los resultados observados se obtienen las siguientes conclusiones:

- La combinación difusor triángulo y concresor Centro de Gravedad es la que obtiene más resultados en Cat5 independientemente del tipo de los conjuntos del universo de entrada y del de salida.
- La combinación difusor singleton y concresor Centro de Gravedad es la que obtiene más resultados en Cat1 para universos de entrada y de salida con conjuntos curvos.
- La combinación difusor singleton y concresor Centro de Gravedad es la que obtiene más resultados si se suman Cat1 + Cat2 + Cat3 para universos de entrada y de salida con conjuntos rectos.
- La combinación difusor singleton y concresor Centro de Gravedad es la que obtiene menos resultados si se suman Cat4 + Cat5 para universos de entrada y de salida con conjuntos rectos.

Por tanto, se concluye que la configuración óptima es: difusor singleton, concresor Centro de Gravedad y universos de entrada y de salida con conjuntos rectos.

4.2.2.3 Pruebas y validación del Agente implementado.

Para comprobar el correcto funcionamiento de la aplicación implementada se somete al sistema a un auto-test consistente en utilizar como consultas de usuario todas las preguntas-tipo definidas en la normalización de los objetos del CCA.

Se realizan los mismos auto-test que se usaron para probar el Método antes de implementar el Agente Inteligente variando los parámetros configurables del Agente con la intención de comparar los resultados obtenidos a fin de determinar la configuración más eficiente.

Los resultados se clasifican según las 5 categorías ya enunciadas en capítulos anteriores y que se enumeran a continuación:

- Cat1. La pregunta correcta es la única que se recupera, o es la que se identifica con el grado de certeza más alto.
- Cat2. La pregunta correcta es la respuesta identificada con el segundo mayor grado de certeza.
- Cat3. La pregunta correcta es la respuesta identificada con el tercer mayor grado de certeza.
- Cat4. La pregunta correcta se identifica como respuesta, pero no entre las tres con mayor grado de certeza.
- Cat5. La pregunta correcta no se encuentra entre las identificadas como respuesta.

Es de recordar que el propósito del Agente no es recuperar únicamente el objeto más afín a la consulta recibida dado que se espera que el usuario pudiera realizar consultas poco precisas o incluso con términos que pudieran inducir a error por lo que el objetivo perseguido al implementar el Agente es que se recuperen los objetos más afines y también los que tengan una relación menos fuerte.

La configuración en el primer auto-test fue la siguiente: Número de entradas 3, número de salidas 1, difusor singleton, concresor Centro de Gravedad, umbrales fijos de valores 0.5, 0.5, y 0.5. Los resultados obtenidos se detallan en la Tabla 17 y muestran un gran rendimiento cuando la representación del objeto está compuesta por entre 2 a 4 términos índices. Cuando la representación del objeto está formada por más de 4 términos-índices, el sistema tiende a considerar que el objeto está relacionado con la consulta aunque no sea así.

Tabla 17: Resultados de los auto-test con variación de parámetros del motor difuso

Test nº	Cat1 (%)	Cat2 (%)	Cat3 (%)	Cat4 (%)	Cat5 (%)
1er. test	43.51	24.22	8.59	11.72	10.16
2º test	54.89	12.03	3.01	0.75	29.32
3er. test	69.93	14.29	3.00	0.75	12.03
4º test	77.44	15.79	4.51	0.75	1.51

La configuración del segundo auto-test realizado es la siguiente: Número de entradas 5, número de salidas 1, difusor singleton, concresor Centro de Gravedad, umbrales fijos de valores 0.5, 0.5, y 0.5. Los resultados obtenidos también se

detallan en la Tabla 17 y en este caso se observa una mejora en los resultados de la Cat1, mejorando por tanto el parámetro precision. Por otro lado, la categoría Cat5 también incrementa el número de resultados lo que significa que el parámetro recall empeora.

Mediante el uso del generador de informes de razonamiento del Agente, se observa al analizar el razonamiento seguido en los casos clasificados en la Cat5, que el fallo en la identificación se produce porque en algún nivel ninguno de los subconjuntos del CCA supera el umbral de aceptación fijado siendo todos rechazados. Este motivo indica claramente que el fallo reside en el algoritmo de determinación por lo que debe ser modificado.

Para ello, se introduce una actuación flexible en relación a los umbrales de aceptación. Si al evaluar los subconjuntos de cualquier nivel todos obtienen certezas inferiores al umbral fijado, se disminuirá dicho valor umbral en 0,05 y se realizará de nuevo la comprobación hasta que alguno supere el nuevo valor umbral. Se aceptarán todos aquellos que superen el nuevo umbral disminuido. Para las determinaciones sucesivas el umbral volverá a tomar el valor inicialmente fijado.

Se repite el auto-test anterior con el algoritmo modificado considerando los umbrales flexibles de valores iniciales 0.5, 0.5, y 0.5 y sin modificar el resto de parámetros. Los resultados de estas pruebas se detallan en la Tabla 17. En los datos se observa que aumentan los resultados de la Cat1, y disminuyen los de la Cat5 por lo que la medida introducida mejora los parámetros recall y precision. Analizando de nuevo los informes de razonamiento de los resultados de la Cat5, se observa claramente que el algoritmo asigna valores de certeza menores cuando en la consulta sólo se encuentran 3 o menos términos-índices. Este análisis lleva a la conclusión de que es conveniente volver a modificar el algoritmo de determinación de forma que cuando en una consulta se identifiquen 3 o menos términos índices se use un motor difuso con 3 entradas para realizar la determinación, mientras que cuando se encuentren más de 3 términos índices se utilice un motor de 5 entradas. Esta modificación está relacionada con el concepto de normalización de los coeficientes del vector representación de un documento del Modelo de Espacio Vectorial (VSM).

Se realiza otro auto-test con la nueva modificación del algoritmo. La configuración utilizada es: Número de entradas 5, número de salidas 1, difusor singleton, concresor Centro de Gravedad, umbrales variables de valores 0.5, 0.5, y 0.5. La Tabla 17 muestra los resultados obtenidos. En ellos se observa una mejora significativa de los valores de ambos estimadores recall y precision alcanzando el 98.49% y el 77.44% respectivamente.

Estos resultados validan la aplicación realizada para implementar el Agente Inteligente para Recuperación de Información basado en los métodos desarrollados en esta tesis.

4.2.3 Conclusiones.

De las pruebas realizadas se concluye que la implementación de un sistema de Recuperación de Información basado en los métodos de IR y TW desarrollados en esta tesis es viable y que alcanza todos los objetivos propuestos.

De forma general, se comprueba también que el motor difuso implementado mantiene la funcionalidad del configurado con matlab y con Unfuzzy utilizado en el desarrollo de los métodos propuestos en esta tesis.

En particular, respecto al método de asignación de coeficientes de peso basado en lógica difusa se comprueba que el conjunto de reglas definidas asigna pesos de valores muy similares a los que asignó previamente el Ingeniero de Conocimiento basándose en su experiencia, y que la nueva asignación no empeora los resultados obtenidos.

Respecto al método de Recuperación de Información se comprueba que es completamente viable su integración en un Agente Inteligente autónomo con el que el usuario pueda interactuar en Lenguaje Natural para realizar una recuperación de información sobre un Conjunto de Conocimiento Acumulado determinado.

4.3 Term Weighting for Information Retrieval Using Fuzzy Logic.

Autores: Jorge Ropero, Ariel Gómez, Alejandro Carrasco, Carlos León, y Joaquín Luque.

DOI: 10.5772/37837

Publicación: Fuzzy Logic - Algorithms, Techniques and Implementations, Prof. Elmer Dadios (Ed.), ISBN: 978-953-51-0393-6, InTech, DOI: 10.5772/2663

Breve descripción: Este capítulo de libro detalla el método de asignación de pesos basado en lógica difusa y abunda en su mejor rendimiento respecto al clásico tf-idf mediante el análisis de las pruebas realizadas. El análisis realizado categoriza las preguntas-tipo utilizadas para la representación de los objetos en base a dos criterios: la naturaleza de la pregunta-tipo utilizada como consulta y el número de preguntas-tipo utilizadas en el proceso de normalización de los objetos.

Índice de calidad: Este libro está citado 4 veces en su conjunto y el capítulo específicamente 1 vez más.

Aportación personal: La aportación del doctorando en esta publicación consiste en la participación en el diseño de las pruebas, en la categorización de las preguntas-tipo, en el análisis de resultados y extracción de conclusiones, y la redacción del artículo.

Term Weighting for Information Retrieval Using Fuzzy Logic.

En esta publicación se realiza un nuevo análisis de los resultados de las pruebas de funcionamiento del método de asignación de pesos basado en lógica difusa desarrollado en esta tesis.

Las pruebas realizadas consisten en usar como consulta de usuario las preguntas-tipo que participan en la normalización de los objetos del CCA y clasificar la respuesta obtenida del sistema en 5 posibles categorías.

Las consultas de usuario se categorizan en base a dos criterios. En concreto:

- Naturaleza de la pregunta-tipo usada como consulta. Esto es, distinguir si la pregunta-tipo utilizada para describir, y posteriormente recuperar, al objeto es una pregunta tipo principal, de sinónimo, imprecisa o específica.
- Número de preguntas-tipo usadas en la representación del objeto. Se categoriza en una sola pregunta-tipo, de 2 a 5, de 6 a 10, y más de 10. El número de preguntas-tipo que participan en la definición de un objeto está asociado en cierta forma con el grado de concreción del concepto representado (del objeto) de forma que cuantas más preguntas-tipo formen parte de la representación del objeto, más difuso es el concepto que representa.

El objetivo perseguido es obtener entre 1 y 5 respuestas entre las que se encuentre la correcta, a poder ser, entre las 3 identificadas con mayor certeza. Se valora, naturalmente, que la respuesta correcta aparezca con el mayor grado de certeza.

Estas pruebas se repiten usando los pesos calculados aplicando el método tf-idf. Los resultados globales de las pruebas se muestran en la Tabla 18 agrupados en las 5 categorías ya definidas en los apartados anteriores.

Tabla 18: Resultados globales de las pruebas categorizadas

Method	Cat1	Cat2	Cat3	Cat4	Cat5	Total
TF-IDF	466	223	53	79	93	914
	(50,98%)	(24,40%)	(5,80%)	(8,64%)	(10,18)	(100%)
FL	710	108	27	28	41	914
	(77,68%)	(11,82%)	(2,95%)	(3,06%)	(4,49%)	(100%)

Según los resultados globales se aprecia que el método tf-idf proporciona valores bastante buenos identificando correctamente el 81.18% de los objetos, y señalando al 50.98% como primera opción (opción más probable). Aún así, el uso del método basado en FL desarrollado consigue aún mejores resultados identificando correctamente el 92.45% de los objetos, y señalando al 77.68% como primera opción (opción más probable).

No obstante, el objetivo perseguido en esta publicación no es realizar un análisis de los resultados globales sino categorizados en base a los dos criterios antes expuestos del tipo de consulta empleada. Con este análisis se pretende deducir cómo se comportará el método en diversas circunstancias, y observar la comparación de rendimiento entre los dos métodos de asignación de pesos utilizados.

4.3.1 Análisis de resultados según la naturaleza de la pregunta-tipo utilizada como consulta.

Para establecer conclusiones sobre los resultados de las pruebas, se analizan los resultados de ambos métodos atendiendo a la clase de pregunta-tipo que se haya utilizado como consulta. Para ello, las preguntas-tipo que representan a los objetos del CCA se clasifican en las 5 categorías siguientes:

- Pregunta-tipo principal. Es una pregunta tipo que define perfectamente el objeto. Añadir otras preguntas-tipo en su representación es opcional.
- Pregunta-tipo de sinónimo. Es una pregunta-tipo que utiliza términos equivalentes a los de otras que definen al mismo objeto. Por ejemplo, sustituir en la misma pregunta-tipo el término informe por documento, memorando, etc.
- Pregunta-tipo imprecisa. Son preguntas-tipo que se crean previendo que el usuario podría no ser experto en la materia sobre la que recaba información y, por tanto, pudiera incluir términos vagos o poco específicos en su consulta. Por ejemplo, sería el caso en el que un usuario preguntara "¿Qué hago con una mesa rota?" en vez de "¿Cómo puedo contactar con el Servicio de Mantenimiento?"
- Pregunta-tipo particular. Se trata de una pregunta-tipo que representa un caso particular del objeto en cuya representación se incluye. Por ejemplo, el usuario podría preguntar sobre los servicios que ofrece la

secretaría virtual (¿Qué servicios ofrece la secretaría virtual de la US?) o preguntar por uno en concreto (¿La Secretaría Virtual ofrece servicio de almacenamiento on cloud?). La naturaleza de esta segunda sentencia sería de tipo particular.

- Preguntas-tipo de usuarios. Son preguntas-tipo realizadas por los usuarios del sistema que inicialmente no se encontraban incluidas en las definiciones de los objetos del CCA pero que se incorporan debido a que el administrador del sistema las considera relevantes por que se repite frecuentemente o por su idoneidad. Suelen proceder de las FAQ del sistema una vez en producción.

La Tabla 19 muestra la clasificación de las preguntas-tipo que participan en las pruebas realizadas.

Tabla 19: Clasificación de las preguntas tipo de las pruebas

Type of standard question	Number of questions
Main standard questions	252
Synonim standard questions.	308
Imprecise standard questions	125
Specific standard questions	229
Feedback standard questions	0
Total standard questions	**914**

Para determinar de donde proviene la mejora detectada en los resultados globales, se analizan los resultados categorizados por la clase de pregunta-tipo utilizada como consulta. Los resultados se muestran en la Tabla 20 y de ellos se concluye que:

Si la pregunta-tipo pertenece a la categoría de “principal”, el método tf-idf se comporta bastante bien identificando el 93.26% entre las 3 primeras categorías (frente al 97.62% del FL) aunque el método basado en FL es más preciso obteniendo un 91.67% de los objetos en la Cat1 frente al 67.86% que obtiene el tf-idf. Este resultado es lógico ya que esta categoría de pregunta-tipo corresponde a una pregunta tipo correctamente formulada.

Si la pregunta-tipo pertenece a la categoría de “sinónimo”, las conclusiones son similares aunque algo peores. Este resultado podría deberse a que el término utilizado como sinónimo no define con la misma claridad al objeto como lo hace el término empleado en la pregunta de categoría “principal”.

Si la pregunta-tipo pertenece a la categoría "imprecisa", las diferencias se acentúan. En el caso del método basado en FL, los resultados se acercan bastante a los obtenidos para las preguntas de categoría "principal" llegando a identificar el 92.80% de los objetos y a señalar el 88.80% de ellos como opción más probable. El método tf-idf también obtiene buenos resultados pero es mucho menos preciso consiguiendo señalar correctamente como opción más probable sólo al 59.20% de los objetos. Este resultado parece indicar que el uso de FL aporta flexibilidad en los términos de la consulta recibida lo que será muy positivo teniendo en cuenta que la consulta realizada por los usuarios reales no coincidirá literalmente con las preguntas-tipo del CCA.

Tabla 20: Resultados categorizados por clase de pregunta tipo

Type of standard question		**Cat1**	**Cat2**	**Cat3**	**Cat4**	**Cat5**	**Total**
Main standard questions	TF-IDF method	171 (67,86%)	58 (23,02%)	6 (2,36%)	6 (2,38%)	11 (4,37%)	252 (100%)
	Fuzzy Logic-based method	231 (91,67%)	13 (5,16%)	2 (0,79%)	0 (0,00%)	6 (2,38%)	252 (100%)
Synonim standard questions	TF-IDF method	177 (57,46%)	86 (27,92%)	13 (4,22%)	15 (4,87%)	17 (5,52%)	308 (100%)
	Fuzzy Logic-based method	252 (81,82%)	41 (13,31%)	3 (0,97%)	5 (1,62%)	47 (2,27%)	308 (100%)
Imprecise standard questions	TF-IDF method	74 (59,20%)	32 (25,60%)	6 (4,80%)	1 (0,80%)	12 (9,60%)	125 (100%)
	Fuzzy Logic-based method	111 (%)88,80	5 (4,00%)	0 (0,00%)	0 (0,00%)	9 (7,20%)	125 (100%)
Specific standard questions	TF-IDF method	46 (20,08%)	49 (21,40%)	26 (11,35%)	55 (24,01%)	52 (22,71%)	229 (100%)
	Fuzzy Logic-based method	107 (46,72%)	53 (23,14%)	24 (10,48%)	23 (10,04%)	22 (9,61%)	229 (100%)

Por último, si la pregunta-tipo pertenece a la categoría de "particular", se obtienen los peores resultados con ambos métodos pero de nuevo el método basado en FL supera al método tf-idf en objetos correctamente identificados. En este caso, el sistema ofrecerá al usuario el objeto más genérico y el objeto más específico debiendo ser dicho usuario el que decida finalmente cual se ajusta mejor a su demanda.

4.3.2 Análisis de resultados según el número de preguntas-tipo utilizadas en la representación del objeto.

Un segundo análisis de resultados basado en categorizar los objetos por el número de preguntas-tipo utilizadas en su representación arroja también resultados interesantes.

En este caso se han considerado el número de preguntas-tipo usadas en la representación del objeto. La idea de partida que motiva este análisis es pensar que un objeto que se define bien utilizando una única pregunta-tipo corresponde a un concepto muy concreto, claro y específico mientras que si es necesario utilizar muchas preguntas-tipo para su representación el objeto corresponderá a un concepto vago, amplio, o impreciso.

En la Tabla 21 se muestran los 4 grupos definidos para el análisis y cuántos objetos participan en cada grupo.

Tabla 21: Preguntas tipo clasificadas por nº de preguntas tipo que participan en la definición del objeto

Group number	Number of standard questions per object	Number of objects
Group 1	1	95
Group 2	2 – 5	108
Group 3	6 – 10	22
Group 4	> 10	28

- Grupo 1: El objeto es definido por una sola pregunta-tipo.
- Grupo 2: Se necesitan de 2 a 5 preguntas-tipo para definir al objeto.
- Grupo 3: Se necesitan de 6 a 10 preguntas-tipo para definir al objeto.
- Grupo 4: Se necesitan más de 10 preguntas-tipo para definir al objeto.

Es evidente que los grupos 1 y 2 son los que contienen más objetos. Parece lógico que no existan muchas preguntas con la misma respuesta. No obstante, los

grupos 3 y 4 también recogen un número importante de objetos cercano al 20% del total, lo que no es nada despreciable.

Para analizar los resultados, se considerará como categoría a la que pertenece la respuesta aquella en la que el objeto obtenga una mayoría de resultados. Esto es, si la representación de un objeto se compone de 15 preguntas-tipo, y al usar cada una de ellas como consulta el sistema devuelve el objeto en segundo lugar (Cat2) en 10 de las ocasiones, se considerará que el objeto se identifica en 2º lugar (Cat2) independientemente del resultado de las otras 5 preguntas-tipo.

Tabla 22: Resultados clasificados por nº de preguntas-tipo de los objetos

Type of standard question		**Cat1**	**Cat2**	**Cat3**	**Cat4**	**Cat5**	**Total**
Group 1	TF-IDF method	74 (77,89%)	16 (16,84%)	1 (1,05%)	1 (1,05%)	3 (3,16%)	95 (100%)
	Fuzzy Logic-based method	89 (93,68%)	3 (3,16%)	2 (2,10%)	0 (0,00%)	1 (1,05%)	95 (100%)
Group 2	TF-IDF method	86 (79,63%)	21 (19,44%)	1 (0,93%)	0 (0,00%)	0 (0,00%)	108 (100%)
	Fuzzy Logic-based method	100 (92,59%)	7 (6,48%)	0 (0,00%)	0 (0,00%)	1 (0,93%)	108 (100%)
Group 3	TF-IDF method	10 (45,45%)	9 (40,91%)	3 (13,63%)	0 (0,00%)	0 (0,00%)	22 (100%)
	Fuzzy Logic-based method	19 (86,36%)	3 (13,63%)	0 (0,00%)	0 (0,00%)	0 (0,00%)	22 (100%)
Group 4	TF-IDF method	10 (35,71%)	10 (35,71%)	3 (10,71%)	2 (7,14%)	3 (10,71%)	28 (100%)
	Fuzzy Logic-based method	21 (75,00%)	4 (14,29%)	1 (3,57%)	1 (3,57%)	1 (3,57%)	28 (100%)

Para los objetos del Grupo 1 y del Grupo 2, los resultados del método basado en FL son casi perfectos y los del método tf-idf también son buenos.

La mayoría de las veces, sobre el 94%, los objetos correctos son identificados por los dos métodos entre los 3 con mayor probabilidad pero el método basado en FL coloca más de ellos en primera posición por lo que ofrece resultados más precisos.

De las pruebas se deduce que los resultados son muy buenos para los dos métodos cuando la representación del objeto tiene hasta 5 preguntas-tipo.

En el caso del Grupo 3 aparece una diferencia significativa entre los resultados de ambos métodos, especialmente en lo referente a la precisión. Mientras que el método basado en FL obtiene un 86.36% de resultados en la Cat1, el tf-idf sólo obtiene el 45.45%.

Esta diferencia es aún mayor cuando el objeto es representado por más de 10 preguntas-tipo. En el caso del Grupo 4 es evidente que ninguna pregunta-tipo es capaz de definir adecuadamente al objeto, lo que indica que la información es vaga e imprecisa, o compleja. En este caso, el método basado en FL identifica correctamente más del 96% de los objetos y es capaz de señalar el 75% como opción más probable mientras que el tf-idf sólo identifica el 82% de los objetos y señala como opción más probable al 35.71%.

De la comparación de los resultados por número de preguntas-tipo que forman la representación del objeto se desprende que el método basado en FL siempre obtiene mejores resultados respecto al método clásico tf-idf pero que es en los casos en los que más preguntas-tipo forman parte de la representación del objeto cuando la diferencia es mayor.

Es de destacar que cuanto más concreto es el concepto que define al objeto, menos preguntas-tipo formarán parte de su representación y que este número irá creciendo conforme el objeto a representar es más complejo o impreciso.

4.3.3 Conclusiones.

De los resultados obtenidos, se concluye que el uso de la FL mejora especialmente los resultados cuando los conceptos a identificar son más complejos, imprecisos, o la información está más desordenada. Esto hace que el método basado en FL desarrollado sea muy adecuado para el uso en IR aplicada al ámbito de los portales web cuyos objetos a menudo corresponden a esta tipología.

Otro aspecto a destacar es que, en los Grupos 3 y 4 de este estudio, se han realizado consultas muy concretas y bien formuladas sobre objetos que representaban conceptos poco específicos, o imprecisos obteniendo los resultados

mostrados. Cabría esperar por tanto que cuando los usuarios poco expertos en la materia hicieran consultas poco específicas o imprecisas sobre objetos pertenecientes a los Grupos 1 y 2, con conceptos claramente definidos y especificados, se obtuvieran resultados similares, otro motivo por el que el uso del método basado en FL contribuye a mejorar el rendimiento del sistema de IR.

5 Resultados de los métodos desarrollados en esta tesis alcanzados en los proyectos de investigación y desarrollo en colaboración con empresas.

5 Resultados de los métodos desarrollados en esta tesis alcanzados en los proyectos de investigación y desarrollo en colaboración con empresas.

La investigación expuesta en esta tesis se ha aplicado, principalmente, en el marco de 3 proyectos de investigación y desarrollo:

6 Conclusiones

6 Conclusiones

En esta tesis por compendio se ha presentado la aplicación de un método de recuperación de información y un método de asignación de pesos, ambos basados en lógica difusa, desarrollados y comprobados en ámbitos de aplicación en portales web, en la construcción de agentes inteligentes para recuperación de información en entornos de ayuda a la docencia y de asistentes virtuales.

También se presenta la implementación de un agente para realizar la aplicación del método de asignación de pesos basado en lógica difusa en los sistemas presentados en el capítulo 5 de esta tesis.

Los resultados de aplicación de ambos métodos se analizan comparativamente con el modelo de espacio vectorial y el método clásico de asignación de pesos tf-idf demostrándose que los métodos basados en lógica difusa desarrollados ofrecen resultados significativamente mejores.

Estos métodos dan lugar a la inscripción en el Registro de la Propiedad Intelectual de Andalucía de un registro de propiedad intelectual, de tipo obra científica.

La aplicación de estos métodos basados en lógica difusa desarrollados en la implementación de un Asistente a la navegación por el portal web de la Universidad de Sevilla, de un Agente Tutor Virtual, y de un Asistente Virtual de acceso y uso de una plataforma de servicios de movilidad, y los resultados de las pruebas de funcionamiento realizadas, demuestran que los estudios realizados en esta tesis pueden integrarse en aplicaciones reales conservando la funcionalidad y capacidades comprobadas en las pruebas de desarrollo de los mismos.

6.1 Futuras líneas de trabajo

Como continuación de la investigación realizada se plantean varias acciones de continuidad:

- Automatización de la clasificación de los objetos del CCA
- Desarrollo del Agente Analizador Gramatical como un módulo independiente que realice el análisis sintáctico completo de una oración en español.
- Estudio sobre datos de consultas reales de los efectos de modificación de:
    - Los valores frontera de los conjuntos difusos
    - La ampliación del rango de la salida proporcionado por el motor difuso
- Estudio de la repercusión sobre la carga computacional de los valores de los umbrales.
- Reestructuración de la base de datos contenedora del CCA del sistema.

7 Referencias

7 Referencias

Este capítulo recoge la bibliografía citada en los artículos que forman parte del compendio y las propias de la memoria de Tesis.

Abulaish, M. & Dey, L., 2005. *Biological ontology enhancement with fuzzy relations: A text-mining framework.* s.l., s.n., pp. 379-385.

Ajayi, A. O., Aderounmu, G. A. & Soriyan, H. A., 2010. An adaptive fuzzy information retrieval model to improve response time perceived by e-commerce clients. *Expert Systems with Applications,* 37(1), pp. 82-91.

Arano, S., 2004. La ontología: una zona de interacción entre la Lingüística y la Documentación. *www. hipertext. net,* Issue 2.

Aronson, A. R. & Rindflesch, T. C., 1997. *Query expansion using the UMLS Metathesaurus.*. s.l., s.n., p. 485.

Aronson, A. R., Rindflesch, T. C. & Browne, A. C., 1994. *Exploiting a Large Thesaurus for Information Retrieval.*. s.l., s.n., pp. 197-216.

Baeza-Yates, R. & Ribeiro-Neto, B., 1999. *Modern information retrieval.* s.l.:ACM press New York.

Belew, R. K., 1989. *Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents.* s.l., s.n., pp. 11-20.

Ben-Dov, M. & Feldman, R., 2005. Text mining and information extraction. En: *Data Mining and Knowledge Discovery Handbook.* s.l.:Springer, pp. 801-831.

Bickmore, T. W., Pfeifer, L. M. & Paasche-Orlow, M. K., 2009. Using computer agents to explain medical documents to patients with low health literacy. *Patient Education and Counseling,* 75(3), pp. 315-320.

Blair, D. C., 1979. Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London: Butterworth; 1979: 208 pp.. *J. Am. Soc. Inf. Sci.,* Nov, 30(6), pp. 374--375.

Cambria, E. & White, B., 2014. Jumping NLP Curves: A Review of Natural Language Processing Research. *IEEE Computational Intelligence Magazine,* 05, 9(2), pp. 48-57.

Chakrabarti, S., 2000. Data mining for hypertext: A tutorial survey. *ACM SIGKDD Explorations Newsletter,* 1(2), pp. 1-11.

Chow, T. W., Zhang, H. & Rahman, M., 2009. A new document representation using term frequency and vectorized graph connectionists with application to document retrieval. *Expert Systems with Applications,* Dec, 36(10), pp. 12023--12035.

Cordón, O., De Moya, F. & Zarco, C., 2004. *Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments.* s.l., s.n., pp. 571-576.

Croft, W. B., 1984. *A comparison of the cosine correlation and the modified probabilistic model.* s.l.:BUTTERWORTH-HEINEMANN LTD THE BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD, OXON, ENGLAND OX5 1GB.

Croft, W. B., 1986. *User-specified domain knowledge for document retrieval.* s.l., s.n., pp. 201-206.

Croft, W. B. & Harper, D. J., 1979. Using probabilistic models of document retrieval without relevance information. *Journal of documentation,* 35(4), pp. 285-295.

Cummins, R. & O'Riordan, C., 2006. Evolved term-weighting schemes in Information Retrieval: an analysis of the solution space. *Artificial Intelligence Review,* 26(1-2), pp. 35-47.

Cuzzocrea, A., 2008. Top-down compression of data cubes in the presence of simultaneous multiple hierarchical range queries. En: *Foundations of Intelligent Systems.* s.l.:Springer, pp. 361-374.

Eisman, E. M., López, V. & Castro, J. L., 2009. Controlling the emotional state of an embodied conversationalagent with a dynamic probabilistic fuzzy rules based system. *Expert Systems with Applications,* 36(6), pp. 9698-9708.

Etzioni, O., 1996. The World-Wide Web: quagmire or gold mine?. *Communications of the ACM,* 39(11), pp. 65-68.

Freitag, D., 1998. *Information extraction from HTML: Application of a general machine learning approach.* s.l., s.n., pp. 517-523.

Friedman, M. y otros, 2004. *A new approach for fuzzy clustering of web documents.* s.l., s.n., pp. 377-381.

Gabora, L., 2013. Toward a theory of creative inklings. *arXiv preprint arXiv:1310.0736.*

García-Serrano, A. M., Martínez, P. & Hernández, J. Z., 2004. Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce. *Expert Systems With Applications,* 26(3), pp. 413-426.

Gehlert, A. & Esswein, W., 2007. Toward a formal research framework for ontological analyses. *Advanced Engineering Informatics,* 21(2), pp. 119-131.

Giaretta, P. & Guarino, N., 1995. Ontologies and knowledge bases towards a terminological clarification. *Towards very large knowledge bases: knowledge building \& knowledge sharing,* Volumen 25, p. 32.

Gómez, A., Ropero, J., León, C. & Carrasco, A., 2008. *A Novel Term Weighting Scheme for a Fuzzy Logic Based Intelligent Web Agent.*. s.l., s.n., pp. 496-499.

Greenes, R. A., 2011. *Clinical decision support: the road ahead.* s.l.:Academic Press.

Haase, V. H., Steinmann, C. & Vejda, S., 2002. Access to knowledge - better use of the internet. *Proceedings of the informing science + IT education conference IS2002.*

Harman, D., 1991. How effective is suffixing?. *JASIS,* 42(1), pp. 7-15.

Hayashi, H. & Yoshida, M., 2004. A Memory Model Based on Dynamical Behavior of the Hippocampus. *Knowledge-Based Intelligent Information and Engineering Systems,* pp. 967--973.

Horng, Y.-J., Chen, S.-M., Chang, Y.-C. & Lee, C.-H., 2005. A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE Transactions on Fuzzy Systems,* Apr, 13(2), pp. 216--228.

Hu, Q. & Huang, J. X., 2010. Passage extraction and result combination for genomics information retrieval. *Journal of Intelligent Information Systems,* 34(3), pp. 249-274.

Iannone, L., Palmisano, I. & Fanizzi, N., 2007. An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence,* 26(2), pp. 139-159.

Johnson, E. & Jones, J., 2008. *A developer's guide to data modeling for SQL server: covering SQL server 2005 and 2008.* s.l.:Addison-Wesley Professional.

Kandel, E. R., 2006. In Search of Memory: The Emergence of a New Science of Mind. *W. W. Norton \& Company; March, 2006; ISBN-13: 978-0393329377.*

Kerly, A., Ellis, R. & Bull, S., 2008. CALMsystem: a conversational agent for learner modelling. *Knowledge-Based Systems,* 21(3), pp. 238-246.

Kim, W., Choi, D. W. & Park, S., 2008. Agent based intelligent search framework for product information using ontology mapping. *Journal of Intelligent Information Systems,* 30(3), pp. 227-247.

Klösgen, W. & Zytkow, J. M., 2002. *Handbook of data mining and knowledge discovery.* s.l.:Oxford University Press, Inc..

Kosala, R. & Blockeel, H., 2000. Web mining research: A survey. *ACM Sigkdd Explorations Newsletter,* 2(1), pp. 1-15.

Kushmerick, N., 2002. *Gleaning answers from the web.* s.l., s.n.

Kwok, K., 1989. *A neural network for probabilistic information retrieval.* s.l., s.n., pp. 21-30.

Laha, A., 2007. Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring. *Advanced Engineering Informatics,* 21(3), pp. 281-291.

Larsen, H. L., 1999. *An approach to flexible information access systems using soft computing.* s.l., s.n., p. 6042.

Larsen, H. L. & Yager, R. R., 1993. The use of fuzzy relational thesauri for classificatory problem solving in information retrieval and expert systems. *Systems, Man and Cybernetics, IEEE Transactions on,* 23(1), pp. 31-41.

Lee, D. L., Chuang, H. & Seamons, K., 1997. Document ranking and the vector-space model. *Software, IEEE,* 14(2), pp. 67-75.

Lertnattee, V. & Theeramunkong, T., 2002. Combining homogeneous classifiers for centroid-based text classification. *Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications.*

Lesk, M. E., 1969. Word-word associations in document retrieval systems. *American documentation,* 20(1), pp. 27-38.

Liao, S.-H., 2005. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications,* 28(1), pp. 93-103.

Liu, D.-R. & Ke, C.-K., 2007. Knowledge support for problem-solving in a production process: A hybrid of knowledge discovery and case-based reasoning. *Expert Systems with Applications,* 33(1), pp. 147-161.

Liu, L. y otros, 2005. A methodical approach to extracting interesting objects from dynamic web pages. *International Journal of Web and Grid Services,* 1(2), pp. 165-195.

Liu, S. y otros, 2001. An approach of multi-hierarchy text classification. *2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No.01EX479).*

Loh, S., de Oliveira, J. P. M. & Gameiro, M. A., 2003. Knowledge Discovery in Texts for Constructing Decision Support Systems. *Applied Intelligence, May 2003, Volume 18, Issue 3, pp 357-366,* 18(3), pp. 357--366.

Luhn, H. P., 1953. A new method of recording and searching information. *American Documentation,* 4(1), pp. 14-16.

Lu, M. y otros, 2002. *SECTCS: towards improving VSM and Naive Bayesian classifier.* s.l., s.n., pp. 5--pp.

Martín Montes, A. & León de Mora, C., 2010. Expert knowledge management based on ontology in a digital library.

Martin, A. & Leon, C., 2009. Intelligent retrieval in a digital library using semantic web. *IADAT Journal of Advanced Technology on Educaction*, 3(3), pp. 427-429.

Mengual, L. y otros, 2001. *Arquitectura multi-agente segura basada en un sistema de implementación automática de protocolos de seguridad.* s.l., s.n.

Mercier, A. & Beigbeder, M., 2005. *Fuzzy proximity ranking with boolean queries.* s.l., s.n.

Mishkin, M., Suzuki, W. A., Gadian, D. G. & Vargha-Khadem, F., 1997. Hierarchical organization of cognitive memory. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 352(1360), pp. 1461-1467.

Moradi, P., Shiri, M. E. & Ebadzadeh, M. M., 2008. *Personalizing results of information retrieval systems using extended fuzzy concept networks.* s.l., s.n., pp. 1-7.

Moscovitch, M., 2008. The hippocampus as a "stupid," domain-specific module: Implications for theories of recent and remote memory, and of imagination.. *Canadian Journal of Experimental Psychology Revue canadienne de psychologie expèrimentale*, 62(1), p. 62.

Olson, D. L. & Shi, Y., 2007. *Introduction to business data mining.* s.l.:McGraw-Hill/Irwin Englewood Cliffs.

Paijmans, J. J., 1999. *Explorations in the document vector model of information retrieval*, s.l.: s.n.

Papadakis, N. K. y otros, 2005. Stavies: A system for information extraction from unknown web data sources through automatic web wrapper generation using clustering techniques. *Knowledge and Data Engineering, IEEE Transactions on*, 17(12), pp. 1638-1652.

Pierre, S., 2002. Intelligent and Heuristic Approaches and Tools for the Topological Design of Data Communication Networks. *Database and Data Communication Network Systems: Techniques and Applications*, Volumen 1, p. 289.

Popovic, M. & Willett, P., 1992. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43(5), pp. 384-390.

Prince, V. & Lafourcade, M., 2003. Mixing semantic networks and conceptual vectors: the case hyperonymy. *The Second IEEE International Conference on Cognitive Informatics, 2003. Proceedings.*.

Quan, T. T., Hui, S. C. & Fong, A. C. M., 2006. Automatic fuzzy ontology generation for semantic help-desk support. *IEEE Transactions on Industrial Informatics*, 2(3), pp. 155-164.

Raghavan, V. V. & Wong, S. M., 1986. A critical analysis of vector space model for information retrieval. *Journal of the American Society for information Science*, 37(5), pp. 279-287.

Rios, S. A., Velasquez, J. D., Yasuda, H. & Aoki, T., 2006. Improving the web site text content by extracting concept-based knowledge. *Lecture Notes in Artificial Intelligence*, 4252(1), pp. 371-378.

Rios, S. y otros, 2006. Improving Web Site Content Using a Concept-Based Knowledge Discovery Process. *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, Dec.

Romero, C. & Ventura, S., 2007. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1), pp. 135-146.

Ropero, J., Gómez, A., León, C. & Carrasco, A., 2007. Information extraction in a set of knowledge using a fuzzy logic based intelligent agent. En: *Computational Science and Its Applications--ICCSA 2007*. s.l.:Springer, pp. 811-820.

Ropero, J., Gómez, A., León, C. & Carrasco, A., 2009. *Term Weighting: Novel Fuzzy Logic based Method Vs. Classical TF-IDF Method for Web Information Extraction.*. s.l., s.n., pp. 130-137.

Ruiz, M. E. & Srinivasan, P., 1998. *Automatic text categorization using neural networks*. s.l., s.n., pp. 59-72.

Ruiz, M. E. & Srinivasan, P., 2002. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1), pp. 87-118.

Salton, G., 1989. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*.

Salton, G. & Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. *Information processing and management*, 24(5), pp. 513-523.

Salton, G., Buckley, C. & Yu, C. T., 1983. An evaluation of term dependence models in information retrieval. *Lecture Notes in Computer Science*, pp. 151--173.

Sato, E., Kawakatsu, J. & Yamaguchi, T., 2004. Networked Intelligent Robots by Ontological Neural Networks. *Knowledge-Based Intelligent Information and Engineering Systems,* pp. 1100--1106.

Sato, N. & Yamaguchi, Y., 2010. Simulation of Human Episodic Memory by Using a Computational Model of the Hippocampus. *Advances in Artificial Intelligence,* Volumen 2010, pp. 1--10.

Sevilla, U. d., 2008. Memoria del Curso Académico 2007-2008. http://servicio.us.es/secgral/normativa/memoria07_08.pdf.

Soliman, M. & Guetl, C., 2013. *Implementing Intelligent Pedagogical Agents in virtual worlds: Tutoring natural science experiments in OpenWonderland.* Berlin, s.n., pp. 782 - 789.

Song, Y.-I.y otros, 2007. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems,* Aug, 31(3), pp. 265--286.

Subasic, P. & Huettner, A., 2001. Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems,* 9(4), pp. 483--496.

Sycara, K. y otros, 1996. Distributed intelligent agents. *IEEE Intelligent Systems,* Issue 6, pp. 36-46.

Tao, Y.-H., Hong, T.-P. & Su, Y.-M., 2008. Web usage mining with intentional browsing data. *Expert Systems with Applications,* Apr, 34(3), pp. 1893--1904.

Thompson, R. H. & Croft, W. B., 1985. An expert system for document retrieval. *Proc. Expert Systems in Government Symposium, pp. 448-456. Washington, DC: IEEE Computer Society Press.*

Webometrics, 2009. http://www.webometrics.info/index.html..

Wik, P. & Hjalmarsson, A., 2009. Embodied conversational agents in computer assisted language learning. *Speech Communication,* Oct, 51(10), pp. 1024--1037.

Wong, S. K. M., Ziarko, W. & Wong, P. C. N., 1985. Generalized vector spaces model in information retrieval. *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR'85.*

Wu, H. & Salton, G., 1981. A comparison of search term weighting. *SIGIR Forum*, Jun, 16(1), pp. 30--39.

Xie, D., 2005. *Fuzzy association rules discovered on effective reduced database algorithm.* s.l., s.n., pp. 779-784.

Xu, J.-S. & Wang, Z.-O., 2003. TCBLSA: a new method of text clustering. *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693).*

Yager, R. R. & Larsen, H. L., 1993. Retrieving information by fuzzification of queries. *Journal of Intelligent Information Systems*, Nov, 2(4), pp. 421--441.

Yu, C. T., Buckley, C., Lam, K. & Salton, G., 1983. A Generalized Term Dependence Model in Information Retrieval. *Information Technology: Research and Development 2:4; 129-154.*

Zadeh, L. A., 1994. Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, Mar, 37(3), pp. 77--84.

Zhai, J., Wang, Q. & Lv, M., 2008. Application of Fuzzy Ontology Framework to Information Retrieval for SCM. *2008 International Symposiums on Information Processing*, May.

Zhang, R. & Zhang, Z., 2003. Addressing CBIR efficiency, effectiveness, and retrieval subjectivity simultaneously. *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval - MIR'03.*

Zhao, Y. & Karypis, G., 2002. Improve precategorized collection retrieval by using supervised term weighting schemes. *Proceedings. International Conference on Information Technology: Coding and Computing.*

Zhou, B. & Yao, Y., 2009. Evaluating information retrieval system performance based on user preference. *Journal of Intelligent Information Systems*, Jun, 34(3), pp. 227--248.

8 Publicaciones científicas

8 Publicaciones científicas.

A continuación se recogen las publicaciones científicas presentadas como aval en el marco de esta tesis por compendio

Expert Systems with Applications 39 (2012) 4567–4581

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A Fuzzy Logic intelligent agent for Information Extraction: Introducing a new Fuzzy Logic-based term weighting scheme

Jorge Ropero *, Ariel Gómez, Alejandro Carrasco, Carlos León

Department of Electronic Technology, University of Seville, Av. Reina Mercedes s/n 41012, Spain

ARTICLE INFO

Keywords:
Information Retrieval
Information Extraction
Fuzzy Logic
Vector Space Model
Index terms
Term weighting
Intelligent agent

ABSTRACT

In this paper, we propose a novel method for Information Extraction (IE) in a set of knowledge in order to answer to user consultations using natural language. The system is based on a Fuzzy Logic engine, which takes advantage of its flexibility for managing sets of accumulated knowledge. These sets may be built in hierarchic levels by a tree structure. The aim of this system is to design and implement an intelligent agent to manage any set of knowledge where information is abundant, vague or imprecise. The method was applied to the case of a major university web portal, University of Seville web portal, which contains a huge amount of information. Besides, we also propose a novel method for term weighting (TW). This method also is based on Fuzzy Logic, and replaces the classical TF–IDF method, usually used for TW, for its flexibility.

1. Introduction

The abundant information due to the rise of Information Technology constitutes an enormous advantage for information searchers. Nevertheless, at the same time, a great problem arises as a result of this increase of data: the difficulty to distinguish the necessary information from the huge quantity of unnecessary data.

For this reason, Information Retrieval (IR) and Information Extraction (IE) have hit the scientific headlines strongly recently. Primarily, both were for document retrieval and extraction, but in the last years its use has been generalized for the search for other types of information, such as the one in a database, a web page or, in general, any set of knowledge. Especially, the so-called Vector Space Model (VSM) is much extended. VSM is based on the use of index terms. These index terms are associated with certain weights, which represent the importance of these terms in the considered set of knowledge. These techniques work reasonably well for IE and IR in many areas, but they have the disadvantage of not being so efficient when user queries are not very specific, or when there is an enormous and heterogeneous amount of information.

In this paper, we propose the development of an intelligent agent that is capable of answering to the needs of the users in their process of retrieving the desired information when it is enormous, heterogeneous, vague, imprecise, or not in order. The main contribution in this paper is the creation of a general method for retrieving and extracting information based on the use of Fuzzy Logic (FL). Fuzzy Logic is an ideal tool for the management of this kind of vague and heterogeneous information. Besides, this method has been implemented and validated for IE in web portals, where the information provided were imprecise and disordered.

Another important contribution in this paper deals with automatic term weighting for VSM. A novel Fuzzy Logic-based TW method is described. This method substitutes the TF-IDF term weighting classic method for its flexibility. In order to show the improvement caused by the new method, some tests have been held on the University of Seville web portal. Moreover, an intelligent agent based on our technology has been developed for the University of Seville and it will be functioning soon. Tests have shown an improvement with a better extraction of the requested information through the new method. Besides, the new method is also better for extracting related information, which might be of interest for users.

This paper has been organized in seven sections. Section 2 constitutes an introduction to IE, IR and Natural Language Processing (NLP). Bearing in mind that we have tested our method for a web portal, concepts like data mining and web mining (WM) are also introduced. Also, VSM is described as the intelligent agent which extracts relevant knowledge is based on it. Since this Agent interacts with users in Natural Language, it is also necessary to introduce the techniques for processing it, comparing the semantic approach with the vectorial one.

Section 3 introduces Fuzzy Logic and the state of the art of FL applications for IE and IR.

In Section 4, our concept of intelligent agent is presented. The analysis of current intelligent agent leads us to considering the major disadvantages derived from the current approach. The reasons for the use of FL for designing intelligent agents are also considered.

* Corresponding author. Tel.: +34 954554325.
E-mail address: jropero@dte.us.es (J. Ropero).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.10.009

Section 4 ends up introducing a FL based general method for knowledge extraction in noisy, disordered and imprecise environments. This method is validated in Section 5, by means of applying it for a web portal, where information has these features.

In Section 6, both TF–IDF TW classic method and a new FL based method are introduced. Besides, both methods are used for tests in the University of Seville web portal. A comparative analysis of the results is made.

Section 7 shows the main conclusions of our work.

2. Information Retrieval and Extraction: Natural Language Processing

The access to the contents of an extensive set of accumulated knowledge – a database, a summary of documents, web contents, goods in a store, pictures, etc – is an important concern nowadays. The users of these data collections may find important difficulties to find the required information. These needs become increased when the information is not in the form of text, the user in question is not habituated the matter, there are ambiguous contents, bad organization or, simply, complex topics or a great amount of information difficult to manage.

Section 2 shows how to find useful information in extensive sets of knowledge and different ways of confronting this problem. Given the need to extract information from the enormous quantity of available information, Section 2.1 introduces data mining, focusing on web mining, as we chose a web portal to validate our IE method. Sections 2.2 and 2.3 approach both IR and IE, respectively. Finally, Section 2.4 is dedicated to NLP, which has a cardinal importance in both tasks.

2.1. Web Mining, WM

Data Mining (DM) is an automatic process of analyzing information in order to discover patterns and to build predictive models (Klogsen & Zytkow, 2002) Applications of DM are numerous covering varied fields: e-commerce, e-learning and educational systems (Romero & Ventura, 2007), financial and marketing applications (Vercellis, 2009; Olson & Shi, 2007), problem solving (Liu & Ke, 2007), biology, medicine and bioengineering (Greenes, 2006), telecommunications (Pierre, 2002) Text Mining (Chakrabarti, 2000; Loh, Palazzo, De Oliveira & Gameiro, 2003) and Web Mining (Pal, Talwar, & Mitra, 2002; Kosala & Blockeel, 2000; Tao, Hong, & Su, 2008).

Nowadays the internet users provide enormous quantities of data sources of text and multimedia. The profusion of resources has caused the need to develop automatic technologies of data mining in the WWW, the so-called web mining (Pal et al., 2002).

Web mining may be divided into four different tasks, as it may be seen in Fig. 1 (Etzioni, 1996): IR, IE, generalization and analysis.

Of these tasks, we focus on Information Retrieval and Information Extraction.

2.2. Information Retrieval

Information Retrieval (IR) is the automatic search of the relevant information contained in a set of knowledge, guaranteeing at the same time that non-relevant retrieved information is as less as possible. The aim must be to reach an improvement in retrieval results according to two key concepts in IR: recall and precision. Recall bears in mind the fact that the most relevant objects for the user must be retrieved. Precision takes into account that strange objects must be rejected. (Ruiz & Srinivasan, 1998). An exact definition of recall and precision is given below.

$$\text{Recall} = \frac{\textit{retrieved relevant objects}}{\textit{total number of relevant objects}} \tag{1}$$

$$\text{Precision} = \frac{\textit{retrieved relevant objects}}{\textit{total number of retrieved objects}} \tag{2}$$

For instance, searching in a collection of 100 documents, in which only 20 are relevant for the user, if the search extracts 18 relevant documents and 7 non relevant ones, recall value is 18/20, that is, 90%, whereas precision value is 18/25 (72%).

IR has been widely used for text classification (Aronson, Ridflesch & Browne, 1994; Liu, Dong, Zhang, Li, & Shi, 2001) introducing approaches such as Vector Space Model (VSM), K nearest neighbor method (KNN), Bayesian classification model, neural networks and Support Vector Machine (SVM) (Lu, Hu, Wu, Lu, & Zhou, 2002). VSM is the most frequently used model. In VSM, a document is conceptually represented by a vector of keywords extracted from the document, with associated weights representing the importance of these keywords in the document. Eventually, these methods have been used not only for text classification but for managing a large amount of information of any kind.

In Vector Space Model (VSM), the content of a document is represented by a vector in a multidimensional space. Then, the corresponding class of the given vector is determined by comparing the distances between vectors. The procedure in VSM may be divided into three stages. The first stage consists of indexing the document, where most relevant terms are extracted from the text of the document. The second stage is based on the introduction of a weight for index terms, in order to improve the search of the relevant content for the user. The last stage classifies the document according to a measure of similarity (Raghavan & Wong, 1986).

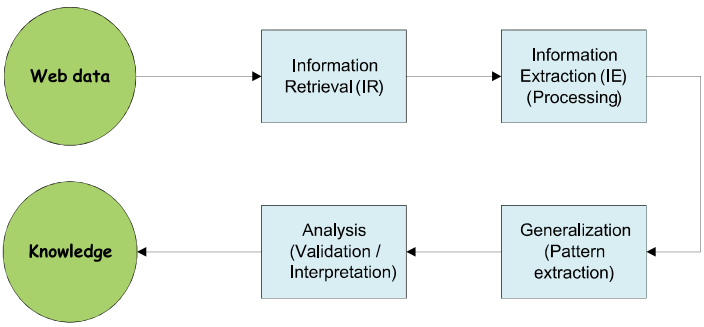


Fig. 1. Web mining tasks.

The most critical stage is the second one, usually called term weighting (TW). Associated weights represent the importance of these keywords in the document. Typically, the so-called TF–IDF method is used for determining the weight of a term (Lee, Chuang & Seamons, 1997). Term Frequency (TF) is the frequency of occurrence of a term in a document and Inverse Document Frequency (IDF) varies inversely with the number of documents to which the term is assigned (Salton, 1988). In Section 6, we discuss the TF–IDF method and we introduce a novel TW Fuzzy Logic based method, which improves the results for Information Extraction.

2.3. Information Extraction

Once documents have been retrieved, the challenge is to extract the required information automatically. Information Extraction (IE) is the task of identifying the specific fragments of a document, which constitute its main semantic content. So far, IE methods involve writing wrappers (Kushmerick, 2002). Some examples of the use of wrappers for IE are STAVIES, which presents a fully automated IE method for web pages (Papadakis, Skoutas, Raftopoulos, & Varvarigou, 2005), or OMINI (Liu, Buttler, Caverlee, Pu, & Zhang, 2005), which introduces tags.

The problem, therefore, is the identification of the fragments of a text that answer to specific questions. Consequently, IE tries to extract new information from the retrieved documents taking advantage of the structure and the representation of the document. Meanwhile, IR experts see the text of a document as a bag of words and do not pay attention to the structure. Scalability is the biggest challenge for IE experts; it is not feasible to build scalable IE systems bearing in mind the size and the dynamism of the web. Therefore, due to the nature of the web, most of IE systems extract information focusing on specific web sites. Other systems use machine learning or data mining techniques for pattern and rule recognition and rules in documents in an automatic or semiautomatic way (Kushmerick, 2002). From this point of view, Web Mining would be part of the Web IE process. The results of this process might be presented in a structured database or as a summary of the texts or original documents.

2.4. Natural Language Processing

Natural Language Processing (NLP) techniques may be used for IR in several ways. As mentioned above, the main aim of using NLP for IR is to improve recall and precision. There are basically two approaches for NLP (Sparck-Jones99), (Aronson & Rindflesch, 1997), (Loh, Palazzo, De Oliveira, & Gameiro, 2003), (Larsen and Yager, 1993), (Berners-Lee and Miller, 2002):

- VSM approach. It is based in the introduction of index terms. An index term may be a keyword (a single word or a word root) or a join term: the latter can be a complex term or a related or similar term. In Fig. 2, different types of index terms are shown.
- Semantic based approach. Though NLP is not an easy task, its potential advantages for IR have made researchers to use both a syntactic approach and a semantic one (Aronson, Rindflesch, & Browne, 1994). It is based on the structure of the considered set of information. A key concept in this field is the concept of ontology (Berners-Lee & Miller, 2002), (Martin & Leon, 2010). Ontology is a common frame or a conceptual automatic and consensual structure to be able to retrieve the required information (Arano, 2003).

Therefore, it is necessary to choose the necessary approach for the web IE system. In the vectorial model, IR and IE are based on the *what* of the information. On the other hand, in semantic webs IR and IE are based on *how* this information is structured. The problem that arises is that, at present, the web does not still provide a great number of ontologies or schemes: only few are and in few matters. Besides, building an ontology from the start turns out to be a hard task and it depends very much on the knowledge engineer who develops it (Iannone, Palmisano, & Fanizzi, 2007). In our research we are inclined for a vectorial approach, though we consider the study of semantic webs a very interesting field of research.

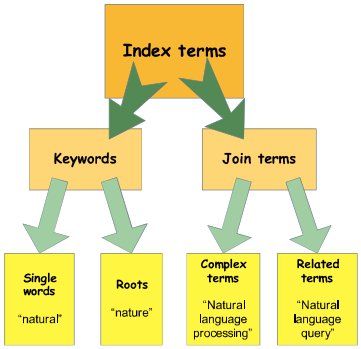


Fig. 2. Different types of index terms.

3. Computational intelligence for knowledge management

3.1. Introduction

It is necessary to consider that the aim of any knowledge access system is to satisfy the needs of the users who access information resources (Larsen, 1999). There are several problems in these knowledge-access systems:

- Information needs are vague or diffuse.
- Information needs change as the user receives this information during his query.
- Users are not conscious of their exact information needs.
- Asking the system about information needs is not usually easy.

Consequently, there is a need to look for a set of methodologies that reflect the notable aptitude of the human being for taking sensible decisions in an imprecise and uncertain environment. This set of methodologies is known as Soft Computing or Computational Intelligence (CI). The main CI tools are Artificial Neural Networks (ANN) and Fuzzy Logic (FL) (Zadeh, 1994).

3.2. Fuzzy Logic applications to knowledge discovery

Search engines, web portals and classic technologies for document retrieval usually consist in searching for keywords in the web. The result may be the finding of thousands of hits, with many of them being irrelevant or maybe not correct or applicable.

There are several approaches at the moment for information handling in an IR system. One of them is based on the Vector Space Model and the other one is related to the concepts of ontology and semantic web. Fig. 3, shows a conceptual scheme on FL applications for IR.

Among VSM based applications for IR, concepts such as queries, clustering, user profiles, and hierarchic relationships take importance (Haase, Steinmann, & Vejda, 2002; Cordon, de Moya, & Zarco, 2004; Mercier & Beigbeder, 2005; Friedman, Last, Zaafrany, Schneider, & Kandel, 2004; Subasic & Huettner, 2001; Horng, Chen,

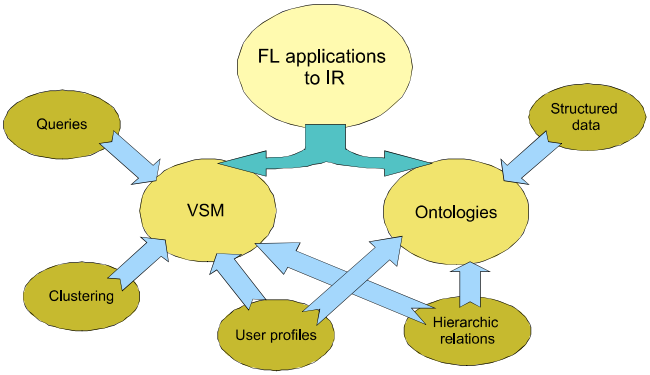


Fig. 3. Conceptual scheme on FL applications for IR.

Chang, & Lee, 2005; Ríos, Velásquez, Yasuda, & Aoki, 2006; Moradi, Ebrahim, & Ebadzadeh, 2008; Zhang & Zhang, 2003). Another possible way of using FL for IR is by means of ontologies. This processing model may help users to have access to the information stored in non-structured or semistructured documents or texts with certain efficiency. Structured data are combined with hierarchic relationships and user profiles for IR applications (Abulaish & Dey, 2005), (Quan, Hui, & Fong, 2006), (Zhai, Wang, & Lv, 2008), (Martin & Leon, 2009).

Anyway, all these applications have something in common with our proposed work on two basic aspects:

- The amount of information is too large to handle.
- The need for a hierarchic structure or the possibility of clustering the information.

Therefore, the ability of FL for the design of an intelligent agent to extract information from a web portal is beyond all doubt.

4. Fuzzy Logic-based intelligent agent

4.1. Intelligent agents for knowledge discovery

The approach to the contents of an extensive set of accumulated knowledge is an important concern nowadays. User needs become increased when the matter is not in the form of text, the user in question is not a habitual user of the matter, there are ambiguous contents, bad organization or, simply, complex topics or a great amount of information difficult to manage (Kwok, 1989). In many cases the solution is to seek some help from an expert on the topic. In fact the person asked to help is an interpreter who is able to generate a syntactically and semantically a correct search obtaining the desired answers. Consequently, there is the need for an agent to interpret the vague information we provide, giving us concrete answers related to the existing contents of the set of knowledge. This should be based on an estimation of the certainty of the relation between what we have expressed in natural language and the contents stored in the set of knowledge (Ropero, Gómez, León, & Carrasco, 2007).

Intelligent Agents, also known as Software Agents, Wizards or Multi-Agent Systems (Turban & Aronson, 2001), are programmed software entities that carry out a series of operations on behalf of a user or another program. They have some degree of independence or autonomy, using some knowledge or representation of the aims or desires of the user (Mengual et al., 2001). If an intelligent agent keeps any kind of conversation with the user, they are also known as Conversational Agents, bots or chatbots (Liao, 2005). At present there are several Intelligent conversational Agents for the most diverse applications, from e-commerce (Ajayi, Aderounmu, & Soriyan, 2009), (Garcia-Serrano, Martinez, & Hernandez, 2004) to virtual education (Kerly, Ellis, & Bull, 2007), (Wik & Hjalmarsson, 2009), or medical uses (Eisman, Lopez, & Castro, 2009), (Bickmore, Pfeifer, & Paasche-Orlow, 2009).

The main problem of most of current agents is, in general, their lack of flexibility. They react well to correct questions, but their answers are far of being too satisfactory when questions are vague or imprecise. And this is the main characteristic when the user is not an expert in the matter – where, in fact, an intelligent agent is more necessary. In addition and also related to this lack of flexibility, many of these agents do not provide more than one answer. It is essential that a user has the possibility of choosing among different chances, as there is a lot of related information in the Internet portals, which might be interesting for the user too.

4.2. Modeling the intelligent agent

4.2.1. Objectives of the intelligent agent

Keeping in mind the limitations of the current intelligent agents, we propose a general IE method using FL for an intelligent agent. An intelligent agent takes advantage of the flexibility the method provides. The method is described in this section. In Section 5 this method is applied to a web portal, using VSM and index terms, based on keywords. As said, above, the information contained in a web page is heterogeneous and vague in most cases, so FL is of great usefulness to find the required information. Besides, we propose a method of consultation based on FL by means of an interface in which it is possible to interact with in NL.

The main objective of the designed system must be to let the users find possible answers to what they are looking for in a huge set of knowledge. With this aim, the whole set of knowledge must be classified into different objects. These objects are the answers to possible user consultations, organized in hierarchic groups. One or more standard questions are assigned for every object, and different index terms from each standard question must then be selected in order to differentiate one object from the others. Finally, term weights are assigned to every index term for every level of hierarchy in a scheme based on VSM. These term weights are the inputs to a FL system. The system must return to the user the objects correspondent with the standard question, or questions that are more similar to the user consultation. The whole process,

together with other concepts defined below, is shown in Fig. 5 (Ropero et al., 2007).

4.2.2. Hierarchic Structure

Provided that the aim of the system is to find the possible answers to user consultations, returning not only the best answer, but also those that are related – user consultations are subject to possible imprecision – it is logical to establish a classification based on a certain criterion or group of criteria. This way, the user might obtain not only the object that is more fitted to his consultation but those that are more closely related.

For instance, in Section 5 we are considering a particular case of IE in a web portal. A hierarchic classification is completely appropriate, since a web portal also has a hierarchic structure. Consequently, it is necessary to identify a web page as an object. That is to say, every web page in a portal is considered to be an object and all these objects are grouped in a hierarchic structure. It is also possible to assign several objects to the same web page if the contained information is heterogenous enough. Likewise, it is necessary to store both the objects and the hierarchic structure of the set of knowledge in databases, as seen in Fig. 4.

4.2.3. Building the intelligent agent

To build the intelligent agent, it is first necessary to bear in mind that user consultations are in Natural Language (NL). We take advantage of this particularity to represent every object as one or several questions in NL, which we have called *standard questions*. Later, it is necessary to extract a series of index terms of the above mentioned standard questions. Finally, term weights must be assigned to these index terms according to the importance of them in the object they are representing. The process consists of two steps:

- The first step is to divide the whole set of knowledge into objects. One or more questions in NL are assigned to every object. The answer to this or these questions must represent the desired object. We have called these questions standard questions. The experience of the Engineer of Knowledge who defines these standard questions as for the jargon in the domain of the set of knowledge is important: the greater his knowledge, the more reliable are the proposed standard questions for the representation of the object. This is due to the fact that they may be more similar to possible user consultations. Nevertheless, it is possible to re-define representations of the object or to add new definitions that should analyze future user consultations and study their syntax and their vocabulary. Consequently, the system can refine his knowledge. In addition, the fact that the intelligent agent is based on FL will provide a greater flexibility.
- The second step is the selection of index terms, which are extracted from standard questions. Index terms represent the most related terms of standard questions with the represented object.

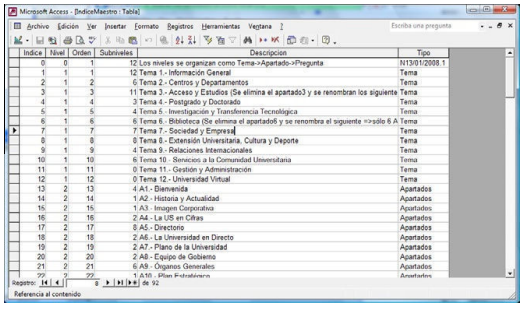


Fig. 4. Database containing the information in a web portal grouped hierarchically.

These index terms may be identified with keywords, but they may be compound terms, too. There exists the need of a series of coefficients associated with index terms whose values must be related somehow to the importance of an term index in the set of knowledge it is representing - it is to say, the importance of the term in every level of the hierarchic structure. These index terms must be stored in a database along with their corresponding weights, corresponding to each of the hierarchic levels. We may consider mainly two methods for term weighting (TW):

- Let an expert in the matter evaluate intuitively the importance of index terms. This method is simple, but it has the disadvantages of depending exclusively on the engineer of the knowledge, it is very subjective and it is not possible to automate.
- Automate TW by means of a series of rules.

Given the large quantity of information there is in a web portal, we choose the second option and we propose a VSM method for TW. The most widely used method for TW is the so-called TF–IDF method. Nevertheless, in this paper we also propose a modification of the method based on the use of FL. This method is described in Section 6. Every index term has an associated weight. This weight has a value between 0 and 1 depending on the importance of the term in every hierarchic level. The greater is the importance of the term in a level, the higher is the weight of the term. In addition, it is necessary to bear in mind that the term weight might not be the same for every hierarchic level, provided that the importance of a word to distinguish, for example, a section from another may be very different from its importance to distinguish between two objects. In short, the whole process of building of the intelligent agent is as summarized in Fig. 5.

For example, a web page may be divided in one o more objects according to the quantity of information it contains. Actually, every object is an answer to every possible user consultation. Since it is possible that several questions drive to the same answer, one or more standard questions may be defined for the same object. Once standard questions are defined, it is necessary to extract the index terms and to assign a weight to them. Index terms and their corresponding weights must be stored in respective databases that constitute a hierarchic structure.

4.3. Mode of operation of the intelligent agent

Once the intelligent agent has been built, it is necessary to know its mode of operation, that is to say, how it works when it receives a user consultation. Index terms are extracted by comparison with the contained ones in their corresponding database. The weights of these index terms for every level constitute the input to an FL system. At this point, the hierarchic structure of the system becomes important. The whole set of knowledge, which constitutes the hierarchic level 0, is divided into level 1 subsets. For each level 1 subset, index terms must have certain weights, which are the possible inputs to an FL engine. The FL engine provides an output for every subset. These outputs are called degrees of certainty. If the degree of certainty corresponding to a subset is lower than a predefined value, named threshold, the content of the corresponding subset is rejected. The aim of using a hierarchic structure is to make possible the rejection of a great amount of content, which will not have to be considered in future queries. For every subset that overcomes the threshold of certainty, the process is repeated. Now, the inputs to the FL engine are the level 2 weights for the corresponding index terms. For the outputs for level 2 subsets, those outputs with a degree of certainty that does not overcome a threshold are rejected

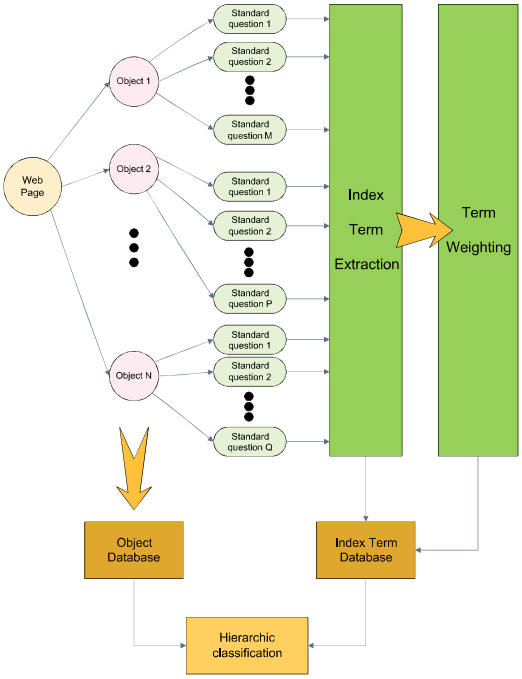


Fig. 5. Process of building an intelligent agent.

again. Otherwise, the process is repeated up to the last level. The final output corresponds to the elements of the last level – that is to say, the objects – whose degree of certainty overcomes the definitive threshold. There is the possibility of several answers. The more vague the queries are, the more answers are obtained. In Fig. 6, the complete process for a two-level hierarchic structure is shown. The whole set of knowledge – level 0 – is grouped in level 1 subsets and these are clustered in level 2 subsets. Since this is the last level, these subsets are own objects.

An application of this methodology for a web portal is described in Section 5 of this paper.

4.4. Fuzzy Logic system

The element of the intelligent agent which determines the degree of certainty for a group of index terms belonging or not to every of the possible subsets of the whole set of knowledge is the fuzzy inference engine. The inference engine has several inputs – the weights of selected index terms – and gives an output – the degree of certainty for a particular subset in a particular level. For the fuzzy engine, it is necessary to define:

- The number of inputs to the inference engine of inference. It depends on the extracted index terms, so it is variable. The inputs are the higher weights of the extracted index terms for every hierarchic level Likewise, it is suitable to define a maximum number of inputs to avoid too vague consultations and, therefore, retrieving too many objects.
- Input fuzzy sets: input ranges, number of fuzzy sets, and shape and range of membership functions.
- Output fuzzy sets: output ranges, number of fuzzy sets, and shape and range of membership functions.

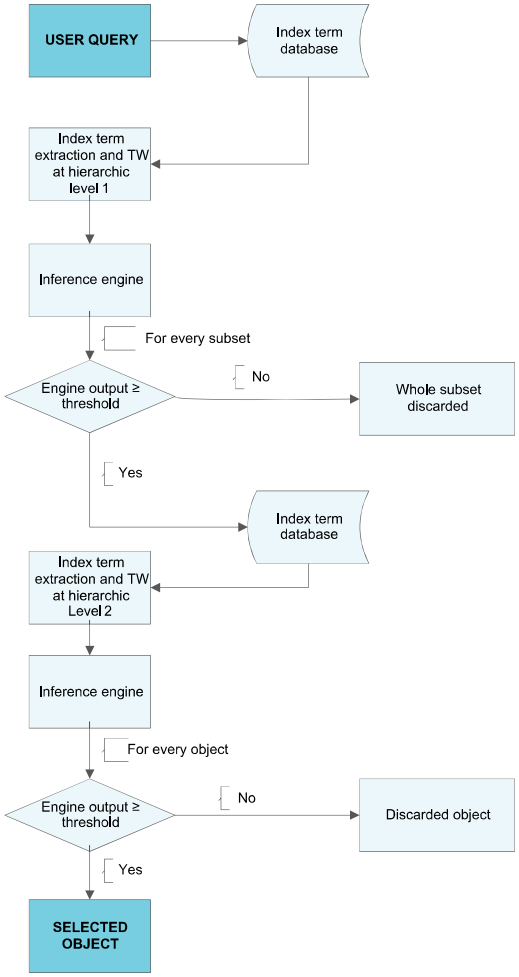


Fig. 6. The complete process of a two-level hierarchic structure.

- Fuzzy rules. They are of the IF ... THEN type. Examples of these rules (81 in total for three inputs) are:
  - IF all inputs are LOW, THEN output is LOW (1 rule for three inputs).
  - IF one input is MEDIUM and the others are LOW, THEN output is MEDIUM-LOW (three rules for three inputs).
  - IF all inputs are MEDIUM or one input is HIGH, THEN output is HIGH (four rules for three inputs).
- Used methods for AND and OR operations and defuzzifying.

All these parameters must be taken into account to find the optimal configuration for the inference engine, core of the intelligent agent. The study of the suitable parameters in the case of a web portal is also described in Section 5.

5. Tests and results

So far, a general method for IR and IE has been proposed. Although we have stood out the method suitability for web applications, this section focuses on the use of this method for IE in web

portals, with the design of an intelligent agent for the web portal of the University of Seville, studying the best parameters for the FL system.

This web portal has 50,000 daily visits, which qualifies it into the 10% most visited university portals and is ranked 223 among more than 4,000 Universities in Webometrics rankings for Universities' web impact (Webometrics., 2009). Moreover, the intelligent agent based on our design is about to start functioning in the University of Seville web portal.

5.1. Set of Knowledge Structure

As the information in the university web portal is abundant, 253 objects grouped in 12 topics were defined. All these groups were made up of a variable number of sections and objects. 2107 standard questions surged from these 253 objects.

As said above, to carry out IE, it is necessary to identify the web page and object, that is to say, every web page in a portal is considered an object. These objects are gathered in a hierarchical structure. Every object is accessible across a unique way of a classification tree. An object is classified under a unique criterion – or group of criteria – (Gómez, Ropero, León, & Carrasco, 2008).

Every object is represented by means of a set of questions, called *standard questions,* formulated in NL. The number of standard questions associated with every web page is variable, depending on the amount of information contained in every page, its importance and the number of index terms synonymous. Logically, system administrator's knowledge about the jargon of the related field is pretty important. The more knowledgeable he is, the higher the reliability of the proposed standard questions becomes, as they shall be more similar to possible user consultations. After all, users are the ones who extract the information.

5.2. Methodology of the intelligent agent

Our study was based both on the study of the web pages themselves and on previous consultations – University of Seville bank of questions. Once standard questions are defined, index terms are extracted from them. We have defined these index terms as words, though there also may be compound terms. Index terms are the ones that better represent a standard question. Every index term is associated with its correspondent term weight. This weight has a value between 0 and 1 and depends on the importance of the term in every hierarchic level. The higher the importance of a term in a level, the higher is the term weight. In addition, term weight is not constant for all levels, as the importance of a word to distinguish a topic from the others may be very different from its importance to distinguish between two objects. An example of the followed methodology is shown in Table 1.

On the other hand, the final aim of the intelligent agent must be to find the Object or Objects whose information is more similar to the requested user consultation. The process that the intelligent agent follows to extract the information related to the user consultation was described in detail in Section 4. To clarify further, we take up the example in Table 2. In the example, a user asks "Which services can I access as a virtual user at the University of Seville?", which corresponds to one of the defined standard questions. We show the process followed by the intelligent agent to extract the requested information and the related one.

Table 1
Example of the followed methodology.

Step	Example
Step 1: Web page identified by standard question/s	– Web page: www.us.es/univirtual/internetwww.us.es/univirtual/internet – Standard question: Which services can I access as a virtual user at the University of Seville?
Step 2: Locate standard question/s in the hierarchic structure.	Topic 12: Virtual University Section 6: Virtual User Object 2
Step 3: Extract index terms	Index terms: 'services', 'virtual', 'user'
Step 4: Term weighting	See Section 6

In this case, the requested information was retrieved, since the user consultation – Which services can I access as a virtual user at the University of Seville? – actually corresponds to a standard question. This standard question refers to Object 12.6.2 (abbreviated notation for the Object corresponding to Topic 12, Section 6, Object 2). In addition, other Objects overcome the defined threshold. Standard questions associated with these Objects are shown in Table 3.

As mentioned above, the first standard question corresponds to the desired Object. In addition, an important advantage is obtained: both the following retrieved standard questions are very much related to the desired Object – they are also related to the Virtual User –. So these Objects may be interesting for the user. The following standard questions are not so similar, but they are somehow related to the query. Our suggestion is to present the web page associated with the first Object to the user and, in another window, among three and five of the following retrieved options.

Moreover, the fact of retrieving other very much related Objects leads us to a conclusion: when the user consultation does not match exactly to any of the stored Objects, the system will try to find the most similar ones. This flexibility is one of the most important advantages of the use of FL.

5.3. Fuzzy Logic engine

As said in Section 4, the core of the intelligent agent is the FL system. For the FL system, we have to consider parameters such as the number of inputs and outputs, fuzzy sets and fuzzy rules.

To prove the efficiency of the proposed system and improve benefits, it was necessary to test the FL system in order to define the suitable parameters for a set of accumulated knowledge. As the portal of the University of Seville has a great amount of information, we tested our method with a more reduced set of knowledge. We used the bank of most frequent questions – answers of the University of Seville. This bank of questions – answers is considered our set of knowledge. It consists of 117 questions, and the results obtained from its use, due to the generality of the method, are applicable to any set of knowledge and, especially, to a web portal.

The first goal of these tests is to check that the system makes a correct identification of standard questions with an index of certainty higher than a certain threshold. The use of Fuzzy Logic makes it possible to identify not only the corresponding standard question but others as well. This is related to the concept of *recall*, though it does not match that exact definition (Ruiz & Srinivasan, 1998). The second goal is to check whether the required standard question is among the three answers with higher degree of certainty. These three answers should be presented to the user. The correct answer must be among these three options. This is related to *precision*, though it does not match that exact definition either.

To do the tests, the so-called standard questions were used as consultations in the Natural Language. The index terms for every standard question must be defined enough to identify the Object related to that standard question. Test results for standard question recognition fit into five categories:

Table 2
FL system response to a user consultation.

Step 1: User query in NL.

Which services can I access as a virtual user at the University of Seville?

Step 2: Index term extraction.

Index term	T1W	T2W	T3W	T4W	T5W	T6W
Services	0.14	0	0	0	0	0.16
User	0	0	0	0	0	0
Virtual	0	0	0.16	0	0	0
Index term	**T7W**	**T8W**	**T9W**	**T10W**	**T11W**	**T12W**
Services	0.16	0	0	0.14	0.16	0.15
User	0	0	0	0.29	0	0.6
Virtual	0	0	0	0	0.16	0.53

TiW = Term Weight Vector for Topic i.

Step 3: Weight vectors are taken as inputs to the fuzzy engine for every topic.

T1O	T2O	T3O	T4O	T5O	T6O
0.29	0.13	0.30	0.13	0.13	0.30
T7O	**T8O**	**T9O**	**T10O**	**T11O**	**T12O**
0.30	0.13	0.13	**0.43**	0.39	**0.62**

TiO = Fuzzy engine output for Topic i.

* Topics 10 and 12 are over the considered threshold – 0.4 in our case.

Step 4: Step 3 is repeated for the next hierarchic level – Sections of the selected Topics.

Index term	T12S1W	T12S2W	T12S3W	T12S4W	T12S5W	T12S6W
Services	0.37	0	0.16	0	0	0.12
User	0	0	0	0	0	0.6
Virtual	0.33	0	0	0.16	0.16	0.45

TiSjW = Term Weight Vector for Topic i, Section j.

T12S1O	T12S2O	T12S3O	T12S4O	T12S5O	T12S6O
0.51	0.13	0.30	0.30	0.30	**0.59**

TiSjO = Fuzzy engine output for Topic i, Section j.

* Topic 10 must also be considered, but we are considering only Topic 12 for simplicity

Step 5: Step 3 is repeated for the next hierarchic level – Objects of the selected Sections.

Index term	T12S6O1W	T12S6O2W	T12S6O3W
Services	0	0.4	0
User	0.57	0.52	0.52
Virtual	0.57	0.52	0.52

TiSjOkW = Term Weight Vector for Topic i, Section j, Object k.

T12S6O1O	T12S6O2O	T12S6O3O
0.6045	**0.7413**	**0.6005**

TiSjOkO = Fuzzy engine output for Topic i, Section j, Object k.

* Topic 12, Section 1 must also be considered, but we are considering only Topic 12, Section 6 for simplicity

1. The correct question is the only one found or the one that has the highest degree of certainty.
2. The correct question is one between the two with the highest certainty or is the one that has the second highest degree of certainty.
3. The correct question is one among the three with the highest degree of certainty or is the one that has the third highest certainty.
4. The correct question is found but not among the three with the highest degree of certainty.
5. The correct question is not found.

These tests are useful to determine the ideal parameters of the FL system for IE. These parameters are described in the following sections.

5.3.1. I/O variables

As said above, the intelligent agent must extract the index terms during a user consultation. The N index terms with a higher weight for evey level of the hierarchy are chosen as inputs to the FL inference engine. Therefore, the first item to do is to determine the suitable number of inputs to the system. The fact of organizing the content hierarchically avoids the need for consulting for the Objects one by one. This is due to the fact that subsets of knowledge whose correspondent output – the output for the FL engine – is lower than a certain threshold are discarded.

In tests, we have considered thresholds of 0.5 for all levels, although these can be modified to obtain better results. In addition, fuzzy sets were defined for inputs and outputs, together with fuzzy rules. The way they were defined is explained in next section.

Table 3
Associated standard questions for the objects retrieved in the example.

Position	Object	Certainty (%)	Associated standard question
1	12.6.2	74.13	Which services can I access as a virtual user at the University of Seville?
2	12.6.1	60.45	I would like to request for an account as a virtual user at the University of Seville
3	12.6.3	60.05	I do not remember my Virtual User password at the University of Seville
4	12.1.5	54.07	I would like to access the Economic Services at the Virtual Secretariat of the University of Seville
5	12.1.6	54.07	I would like to access the management services at the virtual secretariat of the University of Seville
6	10.4.9	48.96	What services does the service of computers and communications offer?
7	12.1.1	41.04	How can I access the virtual secretariat at the University of Seville?

As for the input to the FL system, one to five index terms can be extracted from a consultation. We consider that more than five index terms may not be relevant for IE, so two fuzzy engines were defined: a three-input fuzzy engine and a five-input one. Tests show that using few inputs to a fuzzy engine causes a rapid saturation of the system. This is a great disadvantage for precision: 90% of the correct Objects are detected but only half of them are the first option, as may be seen in Table 4, where results for a three-input fuzzy engine are shown, among the results for other configurations of the engine.

Nevertheless, when a five-input fuzzy engine is used, there are very low values in the degree of certainty. Precision rises to 55%, but recall decreases, as shown in Table 4.

Therefore, we concluded that a low number of inputs affect precision in a negative way, whereas a high number of inputs affect recall. Nevertheless, some improvements may take effect ifa variable number of inputs are used. This point is explained later. In addition, from the analysis of unsuccessful results, it was observed most of the times, the desired Object was not retrieved because the output was below the fixed threshold. There is the possibility of lowering the thresholds of certainty to accept the result as correct. However, this modification takes many erroneous answers as valid, spoiling part of the previous results. The proposed solution is to modify the procedure so that the intelligent agent lowers automatically the fixed threshold only in case that no result overcomes it. With this method, results improve remarkably, as may also be seen in Table 4.

In summary, if the three most probable Objects are retrieved for the user, the desired Object is retrieved 88% of times, and it is the first option 70% of the times.

As for the number of inputs, it is necessary to bear in mind that sometimes it is better to use the three-input fuzzy engine and sometimes the five-input one. We propose a commitment using an input number variable engine dependent on the number of index terms extracted from the user consultation. In the case among one and three extracted index terms, the three-input engine is used. Otherwise, the five-input engine is utilized. Results are shown in Table 4. If the three most probable Objects are retrieved for the user, the desired Object is retrieved 97% of the times, and 77% of the times it is the first option. We consider then that the best choice for I/O parameters is the use of a fuzzy engine with variable output thresholds and a variable number of inputs. This number depends on the number of extracted index terms from a user consultation.

Table 4
Results for different engine configurations. Cat, category.

	Cat1	Cat2	Cat3	Cat4	Cat5
Three-input fuzzy engine results	45%	24%	9%	12%	10%
Five-input fuzzy engine results.	55%	12%	3%	1%	29%
Five-input fuzzy engine results with variable output thresholds.	70%	14%	3%	1%	12%
Five-input fuzzy engine results with variable output thresholds and variable input number fuzzy engine.	77%	16%	4%	1%	2%

5.3.2. I/O Fuzzy set definition

Input range corresponds to weight range for every index term so it is between 0.0 and 1.0. We considered three fuzzy sets represented by the values LOW, MEDIUM and HIGH. Likewise, for simplicity, all of them are kept as triangular although sets were modified later in order to find their ideal shape. The output, which gives the degree of certainty, is also in the 0–1 range, where 0 is the minimum certainty and 1 is the maximum one. Output may be LOW, MEDIUM-LOW, MEDIUM-HIGH and HIGH. These values correspond to output fuzzy sets. The fact that the input takes these three values – LOW, MEDIUM and HIGH – is due to the fact that the number of fuzzy sets is enough so that results are coherent and there are not so many options to let the number of rules increase considerably – in next section this feature is commented, but it seems to be clear that, the higher the number of values, the number of rules defined must be higher. In fact, the outputs were also defined this way – three fuzzy sets – in the beginning, but we introduced one more set for the outputs as we observed a considerable improvement in the tests we made with this modification. The range of values for every input fuzzy set is as follows (Fig. 7):

- LOW, from 0.0 to 0.4 centered in 0.0.
- MEDIUM, from 0.2 to 0.8 centered in 0.5.
- HIGH, from 0.6 to 1.0 centered in 1.0.

The range of values for every output fuzzy set is:

- LOW, from 0.0 to 0.4 centered in 0.0.
- MEDIUM-LOW, from 0.1 to 0.7 centered in 0.4.
- MEDIUM-HIGH, from 0.3 to 0.9 centered in 0.6.
- HIGH, from 0.6 to 1.0 centered in 1.0.

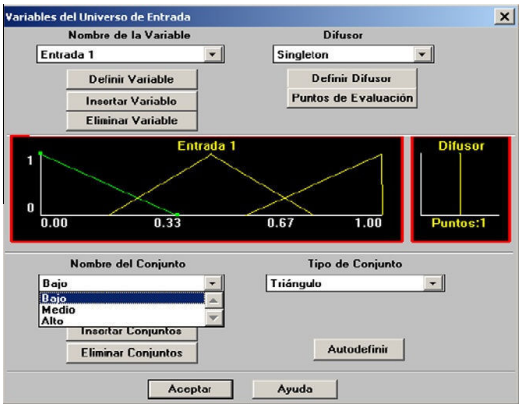


Fig. 7. Fuzzy input set definition.

Table 5
Fuzzy rules for a three input engine.

Rule number	Rule definition	Output
R1	IF one or more inputs = HIGH	HIGH
R2	IF three inputs = MEDIUM	HIGH
R3	IF two inputs = MEDIUM and one input = LOW	MEDIUM–HIGH
R4	IF one input = MEDIUM and two inputs = LOW	MEDIUM–LOW
R5	IF all inputs = LOW	LOW

Eventually, after tests, we came to the conclusion that he best option is the use of:

- Triangular fuzzy sets.
- Singleton fuzzifier.
- Center of gravity defuzzifier.

5.3.3. Rule definition

Once the number of inputs and fuzzy sets has been defined, it is necessary to define the rules for the inference fuzzy engine. Previously, we had established the suitability of implementing a variable input number engine according to the number of index terms extracted from a user consultation. In practice, this causes the implementation of two different inference engines:

- If three or less index terms are extracted, a three input engine is used.
- If more than three index terms are to be extracted, a five input engine is used – if more than five index terms are extracted, only the most significant ones are considered.

Besides, three fuzzy sets had been defined for every input. For a three input engine of three $3^3 = 27$ fuzzy rules were defined, whereas for the five input engine it is necessary to define $3^5 = 243$ rules. This is one of the reasons why more fuzzy sets were not defined: with only one more fuzzy set, it would be necessary define $4^5 = 1024$ rules for the five input engine.

As an example, fuzzy rules defined for the three input engine may be seen in Table 5. These rules cover 27 possible combinations.

6. Fuzzy Logic-based term weighting scheme

Term weighting (TW) is one of the major challenges in IE and IR. The most extended model for IR and IE, as was mentioned in Section 2, is VSM. In VSM, the importance of a term in a subset of knowledge is given by a certain associate weight (Lee et al., 1997). In Section 6.1, there is a brief introduction to TW. In Section 6.2, classic method for TW, so-called TF–IDF is analyzed, and in Section 6.3, we introduce the novel proposed method, based on FL. The values of the weights must be related somehow to the importance of an index term in its corresponding set of knowledge – in our case, Topic, Section or Object. We may consider two options to define these weights:

- An expert in the matter should evaluate intuitively the importance of the index terms: This method is simple, but it has the disadvantage of depending exclusively on the knowledge engineer. It is very subjective and it is not possible to automate the method.
- The generation of automated weights by means of a set of rules: The most widely used method for TW is the TF–IDF method, but we propose a novel Fuzzy Logic based method, which achieves better results in IE.

When a large amount of information needs to be managed, the first option is unfeasible, for it is tedious, dense, and a high level of mastery is necessary on the part of the engineer of knowledge in charge of this task. It is necessary, so, to automate TW.

6.1. The TF–IDF method

Although it was in the late 1950s when the idea of automatic text retrieval systems based on the identification of text content and associated identifiers originated, it was Gerard Salton in the late 1970s and the 80s who laid the foundation for the existing relation between these identifiers and the texts they represent (Salton & Buckley, 1996). Salton suggested that every document D could be represented by term vectors t_k and a set of weights w_{dk}, which represent the weight of the term t_k in document D, that is to say, its importance in the document.

A TW system should improve efficiency in terms of two main factors, recall and precision, as it was mentioned in Section 2. Recall bears in mind the fact that the most relevant objects for the user must be retrieved. Precision takes into account that strange objects must be rejected (Ruiz & Srinivasan, 1998). Recall improves if high-frequency terms are used, because such terms will make it possible to retrieve many objects, including the relevant ones. Precision improves if low-frequency terms are used, as specific terms will isolate the relevant objects from the non-relevant ones. In practice, compromise solutions are used, using terms which are frequent enough to reach a reasonable level of recall without producing a too low precision.

Therefore, terms that are mentioned often in individual objects, seem to be useful to improve recall. This suggests the utilization of a factor named term frequency (TF). Term frequency (TF) is the frequency of occurrence of a term. On the other side, a new factor should be introduced. This factor must favor the terms concentrated in a few documents of the collection. The inverse frequency of document (IDF) varies inversely with the number of objects (n) to which the term is assigned in an N-object collection. A typical IDF factor is log (N/n) (Salton & Buckley, 1996). A usual formula to describe the weight of a term j in document i is given in Eq. 3.

$$W_{ij} = tf_{ij} x idf_j. \tag{3}$$

This formula has been modified and improved by many authors to achieve better results in IR and IE (Lee et al., 1997), (Liu & Ke, 2007), (Zhao & Karypis, 2002), (Lertnattee & Theeramunkong, 2003).

6.2. The FL-based method

The TF-IDF method works reasonably well, but it has the disadvantage of not considering two key aspects for us, as it was explained in ref. (Ropero et al., 2009)

- The first parameter is the degree of identification of the object if only the considered index term is used. This parameter has a strong influence on the final value of a term weight if the degree of identification is high. The more a keyword identifies an object, the higher the value for the corresponding term weight. Nevertheless, this parameter creates two disadvantages in terms of practical aspects when it comes to carrying out a term weight automated and systematic assignment. On the one hand, the degree of identification is not deductible from any characteristic of a keyword, so it must be specified by the System Administrator. The assigned values could be neither univocal nor systematic. On the second hand, the same keyword may have a different relationship with different objects.

- The second parameter is related to join terms. In the index term 'term weighting', this expression would constitute a join term. Every single term in a join term has a lower value than it would have if it did not belong to it. However, if we combine all the single terms in a join term, term weight must be higher. A join term may really determine an object whereas the appearance of only one of its single terms may refer to another object.

The consideration of these two parameters together with classical TF and IDF determines the weight of an index term for every subset in every level. The FL-based method gives a solution to both the problems and also has two main advantages. The solution to both problems is to create a table with all the keywords and their corresponding weights for every object. This table will be created in the phase of keyword extraction from standard questions. Imprecision practically does not affect the working method due to the fact that both term weighting and Information Extraction are based on Fuzzy Logic, which minimizes possible variations of the assigned weights. The way of extracting information also helps to successfully overcome this imprecision. In addition, the two important advantages are the term weighting is automated; and the level of required expertise for an operator is lower. This operator would not need to know anything about the FL engine functioning, but would know only how many times does a term appear in any subset and the answer to these questions:

- Does a keyword undoubtedly define an object by itself?
- Is a keyword tied to another one?

In our case, the application of this method to a web portal, the web portal developer himself may define simultaneously the standard questions and index terms associated with the object – a web page – and the response to the questions mentioned above.

6.3. Implementation of both methods

This section shows how the TF–IDF method and the FL-based method were implemented in practice, in order to compare both methods applying them to the University of Seville web portal.

As mentioned in previous sections, a reasonable measure of the importance of a term may be obtained by means of the product of TF and IDF (TF × IDF). However, this formula has been modified and improved by many authors to achieve better results in IR and IE. Eventually, the chosen formula for our tests was the one proposed by Liu et al. (2001)

$$W_{ik} = \frac{tf_{ik} \times \log(N/n_k + 0.01)}{\sqrt{\sum_{k=1}^{m} tf_{ik} \times \log(N/n_k + 0.01)^2}} \quad (4)$$

Here tf_{ik} is the ith term frequency of occurrence in the kth subset – Topic/Section/Object – n_i is the number of subsets to which the term T_i is assigned in a collection of N objects. Consequently, it is taken into account that a term might be present in other sets of the collection.As an example, we are using the term 'virtual', as used in the example in Section 5.

At Topic level:

- 'Virtual' appears 8 times in Topic 12 ($tf_{ik} = 8$, $K = 12$).
- 'Virtual' appears twice in other Topics ($n_k = 3$)
- There are 12 Topics in total ($N = 12$) – for normalizing, it is only necessary to know the other tf_{ik} and n_k for the Topic –.
- Substituting, $W_{ik} = 0.20$.

At Section level:

- 'Virtual' appears 3 times in Section 12.6 ($tf_{ik} = 3$, $K = 6$)
- 'Virtual' appears 5 times in other Sections in Topic 12 ($n_k = 6$)
- There are 6 Sections in Topic 12 ($N = 6$).
- Substituting, $W_{ik} = 0.17$.

At Object level:

- 'Virtual' appears once in Object 12.6.2 ($tf_{ik} = 1$, $K = 2$). – Logically a term can only appear once in an Object –.
- 'Virtual' appears twice in other Topics ($n_k = 3$)
- There are 3 Objects in Section 12.6 ($N = 3$).
- Substituting, $W_{ik} = 0.01$. In fact, 'virtual' appears in all the Objects in Section 12.6, so it is irrelevant to distinguish the Object.

Consequently, 'virtual' would be relevant to find out that the Object is in Topic 12, Section 6, but irrelevant to find out the definite Object, which should be found according to other terms in a user consultation.

However, TF-IDF has the disadvantage of not considering the degree of identification of the object if only the considered index term is used and the existence of tied keywords. Like TF-IDF method, it is necessary to know TF and IDF, and also the answer to the questions mentioned above. FL-based term weighting method is defined below. Four questions must be answered to determine the Term Weight of an Index Term:

- Question 1 (Q1): How often does an index term appear in other subsets? – Related to IDF.
- Question 2 (Q2): How often does an index term appear in its own subset? – Related to TF.
- Question 3 (Q3): Does an index term undoubtedly define an object by itself?
- Question 4 (Q4): Is an index term tied to another one?

The answer to these questions gives a series of values which are the inputs to a Fuzzy Logic system, called Weight Assigner. The output of the Weight Assigner is the definite weight for the correspondent index term. The followed scheme may be observed in Fig. 8.

Subsequently, the way of defining input values associated with each of four questions is described.

6.3.1. Question 1

Term weight is partly associated with the question 'How often does an index term appear in other subsets?'. It is given by a value between 0 – if it appears many times – and 1 – if it does not appear in any other subset. To define weights, we are considering the times that the most used terms in the whole set of knowledge appear. The list of the most used index terms is as follows:

1. Service: 31 times.
2. Services: 18 times.
3. Library: 16 times.
4. Research: 15 times.
5. Address: 14 times.

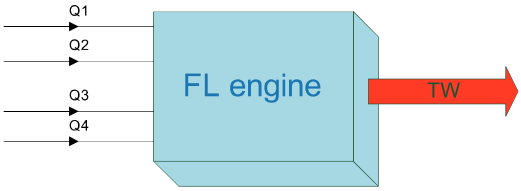


Fig. 8. TW generation method scheme.

+. Student: 14 times
7. Mail: 13 times.
+. Access: 13 times.
9. Electronic: 12 times.
+. Computer: 12 times.
+. Resources: 12 times.
12. Center: 10 times.
+. Education: 10 times.
+. Registration: 10 times.
+. Program: 10 times.

Provided that there are 1114 index terms defined in our case, we think that 1% of these words must mark the border for the value 0 (11th word). Therefore, whenever an index term appears more than 12 times in other subsets, we will give it the value of 0. Values for every Topic are defined in Table 6. Appearing between 0 and 3 times – approximately a third of the possible values – we consider that an index term belongs to the so-called HIGH set. Therefore, it is defined in its correspondent fuzzy set with uniformly distributed values between 0.7 and 1. Analogously, we may distribute all values uniformly according to different fuzzy sets. Fuzzy sets are triangular, on one hand for simplicity and on the other hand because we tested other more complex types of sets (Gauss, Pi type, etc) and the results did not improve at all.

Provided that different weights are defined in every hierarchic level, we should consider other scales to calculate them. As for the Level Topic we were considering the immediately top level – the whole set of knowledge; for the Section level we should consider the times that an index term appears in a certain Topic. The list of the most used index terms in a unique Topic is the following:

1. Service: Topic 10, 16 times.
2. Address: Topic 1, 10 times.
+. Library: Topic 6, 10 times.
+. Registration: Topic 3, 10 times.
5. Mail: Topic 1, 9 times.
+. Electronic: Topic 1, 9 times.
7. Virtual: Topic 12, 8 times.
8. Computer: Topic 10, 7 times.
+. Services: Topic 1, 7 times.
10. Education: Topic 1, 5 times.
+. Resources: Topic 12, 5 times.

In the same way, at the level of Topic, term weight has a value between 0 – if it appears many times – and 1 – if it does not appear in any other subset. We again consider that 1% of these words must mark the border for the value 0 - 11 words – so whenever a term appears more than 5 times in other subsets, its weight takes the value 0 at the Section level.

Possible term weights for the level of Section are shown in Table 7. The method is analogous and considers the definition of the fuzzy sets. At the level of Object, term weights are shown in Table 8.

6.3.2. Question 2

To find out the term weight associated with question 2 – Q2. How often does an index term appear in its own subset? – the reasoning is analogous. However, we have to bear in mind that it is necessary to consider the frequency inside a unique set of knowledge, thus the number of appearances of index terms decreases considerably. The list of the most used index terms in a Topic must be considered again. It also must be born in mind that the more an index term appears in a Topic or Section, the higher the value for an index term is. Q2 is senseless at the level of Object. The proposed values are given in Table 9.

6.3.3. Question 3

In the case of question 3 – Q3. Does a term define undoubtedly a standard question? – the answer is completely subjective and we propose the answers 'Yes', 'Rather' and 'No'. Term weight values for this question are shown in Table 10.

6.3.4. Question 4

Finally, question 4 – Q4. Is an index term tied to another one? – deals with the number of index terms tied to another one. We propose term weight values for this question in Table 11. Again, the values 0.7 and 0.3 are a consequence of considering the border between fuzzy sets.

After considering all these factors, fuzzy rules for Topic and Section levels are defined in Table 12. These rules cover all the 81 possible combinations. Note that, apart from the three input sets mentioned in previous sections, four output sets have been defined - HIGH, MEDIUM-HIGH, MEDIUM-LOW and LOW. At the level of Object, we must discard question 2 and rules change.

The only aspect which has not been defined yet is multiple appearances in a Topic or Section. For example, it is possible that

Table 6
Term weight values for every Topic for Q1.

Times appearing	0	1	2	3	4	5	6	7	8	9	10	11	12	⩾13
Value	1	0.9	0.8	0.7	0.64	0.59	0.53	0.47	0.41	0.36	0.3	0.2	0.1	0

Table 7
Term weight values for every Section for Q1.

Times appearing	0	1	2	3	4	5	⩾6
Value	1	0.7	0.6	0.5	0.4	0.3	0

Table 8
Term weight values for every Object for Q1.

Times appearing	0	1	2	⩾3
Value	1	0.7	0.3	0

Table 9
Term weight values for every Topic and Section for Q2.

Times appearing	0	1	2	3	4	5	⩾6
Value	1	0.7	0.6	0.5	0.4	0.3	0

Table 10
Term weight values for Q3.

Answer to Q3: Does a term define undoubtedly a standard question?	Yes	Rather	No
Value	1	0.5	0

Table 11
Term weight values for Q4.

Number of index terms tied to another index term	0	1	2	⩾3
Value	1	0.7	0.3	0

Table 12
Rule definition for Topic and Section levels.

Rule number	Rule definition	Output
R1	IF Q1 = HIGH and Q2 ≠ LOW	At least MEDIUM–HIGH
R2	IF Q1 = MEDIUM and Q2 = HIGH	At least MEDIUM–HIGH
R3	IF Q1 = HIGH and Q2 = LOW	Depends on other Questions
R4	IF Q1 = HIGH and Q2 = LOW	Depends on other Questions
R5	IF Q3 = HIGH	At least MEDIUM–HIGH
R6	IF Q4 = LOW	Descends a level
R7	IF Q4 = MEDIUM	If the Output is MEDIUM–LOW, it descends to LOW
R8	IF (R1 and R2) or (R1 and R5) or (R2 and R5)	HIGH
R9	In any other case	MEDIUM–LOW

the answer to question 3 is 'Rather' in one case 'No' in another one. In this case, a weighted average of the corresponding term weights is calculated.

An example of all the processes is shown below

Example.

Object 12.6.2 is defined by the following standard question:

Which services can I access as a virtual user at the University of Seville?

If we consider the term 'virtual':

- At Topic level:
- 'Virtual' appears twice in other Topics in the whole set of knowledge, so that the value associated with Q1 is 0.80.
- 'Virtual' appears 8 times in Topic 12, so that the value associated with Q2 is 1.
- The response to Q3 is 'Rather' in 5 of the 8 times and 'No' in the other three, so that the value associated with Q3 is a weighted average: $(5*0.5 + 3*0)/8 = 0.375$.
- Term 'virtual' is tied to one term 7 times and it is tied to two terms once. Therefore, the average is 1.14 terms. A linear extrapolation leads to a value associated with Q4 of 0.65.
- With all the values as inputs for the Fuzzy Logic engine, we obtain a term weight of 0.53.
- At Section level:
- 'Virtual' appears 5 times in other Sections corresponding to Topic 12, so that the value associated with Q1 is 0.30.
- 'Virtual' appears 3 times in Topic 12, so that the value associated with Q2 is 0.45.
- The response to Q3 is 'Rather' in all cases, so that the value associated with Q3 is 0.5.
- Term 'virtual' is tied to term 'user' so that the value associated with Q4 is 0.7.
- With all the values as inputs for the Fuzzy Logic engine, we obtain a term weight of 0.45.
- At Object level:
- 'Virtual' appears twice in other Objects corresponding to Section 12.6, so that the value associated with Q1 is 0.30.
- The response to Q3 is 'Rather', so that the value associated to Q3 is 0.5.
- Term 'virtual' is tied to term 'user' so that the value associated with Q4 is 0.7.
- With all the values as inputs for the Fuzzy Logic engine, we obtain a term weight of 0.52. We can see the difference with the corresponding term weight obtained with the TF-IDF method, but it is exactly what we are looking for: not only the desired object but the most closely related to it must be retrieved.

To compare results, we considered the position in which the correct answer appeared among the retrieved answers, according to fuzzy engine outputs. For it, the first necessary step to follow is to define the overcoming thresholds for the fuzzy engine. This way, Topics and Sections that are not related with the Object to identify are eliminated. We also have to define low enough thresholds, in order to be able to obtain related Objects also. We suggest presenting between 1 and 5 answers, depending on the number of related Objects. As explained in previous sections, term weights are lower for TF-IDF method, due to normalization. For this reason, thresholds were fixed to 0.2 to overcome the level of Topic and 0.3 to overcome the level of Section for the method TF–IDF. Meanwhile, both thresholds have a value of 0.4 for the FL-based method.

Table 13
Comparison between TF-IDF classic method and the novel FL-based method.

	Cat1	Cat2	Cat3	Cat4	Cat5	Total
TF-IDF Method	466 (50.98%)	223 (24.40%)	53 (5.80%)	79 (8.64%)	93 (10.18%)	914
FL Method	710 (77.68%)	108 (11.82%)	27 (2.95%)	28 (3.06%)	41 (4.49%)	914

The results of the consultation were sorted in 5 categories:

- Category Cat1: the correct answer is retrieved as the only answer or it is the one that has a higher degree of certainty among the answers retrieved by the system.
- Category Cat2: The correct answer is retrieved between the 3 with a higher degree of certainty -excluding the previous case.
- Category Cat3: The correct answer is retrieved among the 5 with a higher degree of certainty - excluding the previous cases.
- Category Cat4: The correct answer is retrieved, but not among the 5 with a higher degree of certainty.
- Category Cat5: The correct answer is not retrieved by the system.

The ideal situation comes when the desired Object is retrieved as Cat1, though Cat2 and Cat3 would be reasonably acceptable. The results obtained in the tests are shown in Table 13. Though the obtained results with the TF-IDF method are quite reasonable, 81.18% of the objects being retrieved among the first 5 options - and more than as Cat1, the FL based method turns out to be clearly better, with 92.45% of the desired Objects retrieved - and more than three quarters as the first option.

More detailed tests were made, according to the type of standard questions and the number of standard questions defined for every Object. We came to the conclusion that the more intricate, disordered and confused the information is, the better the FL TW method is, compared with the classic TF-IDF one. This makes its application ideal for the case of an intelligent agent for a web portal, where the information has these features and users may carry out inaccurate or disoriented consultations.

7. Conclusions

In this paper, we present a novel general method for IE and IR by means of the use of an intelligent agent based on FL. Given the lack of flexibility of most of intelligent agents when information is abundant, confused, vague or heterogeneous, we propose an IE method based on the VSM and FL. A set of knowledge is divided in different hierarchic levels up to a level where the instances or Objects are extracted. A series of standard questions are assigned to every Object, based on the possible consultations from a user in Natural Language. These questions drive to the extraction of index terms.

Index terms have associated term weights, according to their importance in their correspondent subset of knowledge. Given

Fig. 9. Prototype of the intelligent agent developed for University of Seville.

the need to automate TW we propose a novel TW method based on the use FL. This method replaces the classic method, the so-called TF-IDF method.

This method has been applied in development of the University of Seville intelligent agent, which is to be functioning soon. An image of the prototype is shown in Fig. 9.

We also propose some future lines of investigation. First of all, the study of the ontology based instead of the vectorial approach. The fact that there is difficulty in using ontologies does not mean that we should not consider this quite an interesting field of investigation. Secondly, CI techniques, other than FL, can be applied to build intelligent agents. Specifically, neuro-fuzzy techniques are a very interesting possibility, as they combine the human reasoning of FL with the neural connection based structure of the ANN, taking advantage of both techniques.

References

Abulaish, M., & Dey, L. (2005). Biological ontology enhancement with fuzzy relations: A text-mining framework. In *Proceedings of the 2005 IEEE/WIC/ACM international conference on web intelligence, France* (pp. 379–385).

Ajayi, A. O., Aderounmu, G. A., & Soriyan, H. A. (2009). An adaptive fuzzy Information Retrieval model to improve response time perceived by e-commerce clients. *Expert Systems with Applications (ESWA), 37*(1), 82–91.

Arano, S. (2003). La ontología: una zona de interacción entre la Lingüística y la Documentación. *Hipertext.net*, No. 2.

Aronson, A. R., & Rindflesch, T. C. (1997). Query expansion using the UMLS Metathesaurus. In *Proceedings of the 1997 AMIA Annual Fall Symposium* (pp. 485–489).

Aronson, A. R., Rindflesch, T. C., & Browne, A. C. (1994). Exploiting a large thesaurus for information retrieval. In *Proceedings of RIAO* (pp. 197–216).

Berners-Lee, T., & Miller, E (2002). The Semantic Web lifts off. *ERCIM News No. 51. Special Semantic Web.*

Bickmore, T. W., Pfeifer, L. M., & Paasche-Orlow, M. K. (2009). Using computer agents to explain medical documents to patients with low health literacy. *Patient Education and Counseling, 75*(3), 315–320.

Chakrabarti, S. (2000). Data Mining for hypertext: A tutorial survey. *ACM SIGKDD Explorations, Newsletter of the Special Interest Group on Knowledge Discovery and Data Mining.*.

Cordon, O., de Moya, F., & Zarco, C. (2004). Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments. In *Proceedings of the IEEE international conference on fuzzy systems* (Vol. 1, pp. 571–576).

Eisman, E. M., Lopez, V., & Castro, J. L. (2009). Controlling the emotional state of an embodied conversationalagent with a dynamic probabilistic fuzzy rules based system. *Expert Systems with Applications, 36*(6), 9698–9708.

Etzioni, O. (1996). The World Wide Web: Quagmire or gold mine. *Communicational ACM, 39*(11), 65–68.

Friedman, M., Last, M., Zaafrany O., Schneider, M., & Kandel, A. (2004). A new approach for fuzzy clustering of web documents. In *Proceedings of the IEEE international conference on fuzzy systems* (Vol. 1, pp. 377–381).

Garcia-Serrano, A., Martinez, P., & Hernandez, J. (2004). Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce". *Expert Systems with Applications, 26*(3), 413–426.

Gómez, A., Ropero, J., León, C., & Carrasco, A. (2008). A novel term weighting scheme for a fuzzy logic based intelligent web agent. In *ICEIS 2008 – Proceedings of the 10th international conference on enterprise information systems, AIDSS* (pp. 496–499).

Greenes, R. A. (2006). *Clinical decision making - The road ahead*. Elsevier.

Haase, V. H., Steinmann, C., & Vejda, S. (2002). Access to knowledge: better use of the internet. In *IS2002 Proceedings of the informing science + IT education conference, Cork, Ireland* (pp. 618–627).

Horng, Y. J., Chen, S. M., Chang, Y. C., & Lee, C. H. (2005). A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE T. Fuzzy Systems, 2*, 216–228. 13.

Iannone, L., Palmisano, I., & Fanizzi, N. (2007). An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence, 26*(2), 139–159, ISSN 0924-669X 2007, Springer.

Kerly, A., Ellis, R., & Bull, S. (2007). CALMsystem: A Conversational Agent for Learner Modelling. *Knowledge-Based Systems, 21*(3), 238–246.

Klogsen, W., & Zytkow, J. (2002). *Handbook of data mining and knowledge discovery*. New York: Oxford University Press.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. , *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining*. ACM (Vol. 2).

Kushmerick, N. (2002). Gleaning answers from the web. In *Proceedings of the AAAI spring symposium on mining answers from texts and knowledge bases, Palo Alto* (pp. 43–45).

Kwok, K. L. (1989). A neural network for probabilistic information retrieval. In *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval, Cambridge, Massachusetts, United States*

Larsen, H. L. (1999). An approach to flexible information access systems using soft computing. In *Proceedings of the 32nd annual Hawaii international conference on system sciences, HICCS99* (pp. 5–8).

Larsen, H., & Yager, R. (1993). The use of fuzzy relational thesauri for classificatory problem solving in information retrieval and expert systems. In *Proceedings of the IEEE transactions on systems, man, and cybernetics* (Vol. 23, pp. 31–41).

Lertnattee, V, & Theeramunkong, T. (2003). Combining homogenous classifiers for centroid-based text classification. In *Proceedings of the 7th international symposium on computers and communications* (pp. 1034–1039).

Lee, D. L., Chuang, H., & Seamons, K. (1997). Document ranking and the vector-space model. In *IEEE Software* (Vol. 14, pp. 67-75).

Liao, S. H. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications, 28*(1), 93–103.

Liu, D. R., & Ke, C. K. (2007). Knowledge support for problem-solving in a production process: A hybrid of knowledge discovery and case-based reasoning. *Expert Systems with Applications., 33*(1), 147–161.

Liu, L., Buttler, D., Caverlee, J., Pu, C., & Zhang, J. (2005). A methodical approach to extracting interesting objects from dynamic web pages. *International Journal of Web and Grid Services, 1*(2), 165–195.

Liu, S., Dong, M., Zhang, H., Li, R., & Shi, Z. (2001). An approach of multi-hierarchy text classification. In *Proceedings of the international conferences on info-tech and info-net, Beijing* (Vol. 3, pp. 95–100).

Loh, S., Palazzo, J., De Oliveira, M., & Gameiro, M. (2003). Knowledge discovery in texts for constructing decision support systems. *Applied Intelligence, New York, NY, USA, 18*(3), 357–366.

Lu, M., Hu, K., Wu, Y., Lu, Y., & Zhou, L. (2002). SECTCS: towards improving VSM and Naive Bayesian classifier. *IEEE International Conference on Systems, Man and Cybernetics, 5*, 5.

Martin, A., & Leon, C. (2009). Intelligent retrieval in a digital library using semantic web. *IADAT Journal of Advanced Technology on Educaction, 3*(3), 427–429.

Martin, A., & Leon, C. (2010). Expert knowledge management based on ontology in a digital library. In *ICEIS 2010 12th international conference on enterprise information systems, Madeira (Portugal)* (pp. 291–298).

Mengual, L., Barcia, N., Bobadilla, J. Jimenez, R., Setien, J., & Yaguez, J. (2001). Arquitectura multi-agente segura basada en un sistema de implementación

automática de protocolos de seguridad. I Simposio Español de Negocio Electrónico.
Mercier, A., & Beigbeder, M. (2005). Fuzzy proximity ranking with Boolean queries. In *Proceedings of the 14th text retrieval conference (TREC), Gaithersburg, Maryland, USA*.
Moradi, P., Ebrahim, M., & Ebadzadeh, M. M. (2008). Personalizing results of information retrieval systems using extended fuzzy concept networks. In *3rd International conference on information and communication technologies: From theory to applications, ICTTA* (pp. 1–7).
Olson, D., & Shi, Y. (2007). *Introduction to business data mining*. McGraw-Hill.
Pal, S. K., Talwar, V., & Mitra, P. (2002). Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. *IEEE Transactions on Neural Networks, 13*(5), 1163–1177.
Papadakis, N. K., Skoutas, D., Raftopoulos, K., & Varvarigou, T. A. (2005). STAVIES: A System for Information Extraction from Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques. *IEEE Transactions on Knowledge and Data Engineering, 17*(12), 1638–1652.
Pierre, S. (2002). *Intelligent and Heuristic Approaches and Tools for the Topological Design of Data Communication Networks. Data Communication Network Techniques and Applications*. New York: Academic Press, pp. 289–326.
Quan, T. T., Hui, S. C., & Fong, A. C. M. (2006). Automatic fuzzy ontology generation for semantic help-desk support. *Industrial Informatics, IEEE Transactions on, 2*(3), 155–164.
Raghavan, V. V., & Wong, S. K. (1986). A critical analysis of Vector Space Model for information retrieval. *Journal of the American Society for Information Science, 37*(5), 279–287.
Ríos, S. A., Velásquez, J. D., Yasuda, H., & Aoki, T. (2006). Improving the web site text content by extracting concept-based knowledge. *Lecture Notes in Artificial Intelligence, 1*, 371–378. 4252.
Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications., 33*(1), 135–146.
Ropero, J., Gómez, A., León, C., & Carrasco, A. (2007). *Information Extraction in a set of knowledge using a Fuzzy Logic based intelligent agent. Lecture Notes in Computer Science* (Vol. 47). LNCS, part 3, pp. 811–820.
Ropero, J., Gomez, A., Leon, C., & Carrasco, A (2009). Term weighting: Novel fuzzy logic based method Vs. classical TF-IDF method for Web information extraction. *Proceedings of the 11th international conference on enterprise information systems* (pp. 130-137).
Ruiz, M., & Srinivasan, P. (1998). Automatic text categorization using neural networks . In E. Efthimiadis (Ed.), *Advances in classification research, vol. 8: Proceedings of the 8th ASIS SIG/CR classification research workshop*. New Jersey: Information Today, Medford (pp. 59–72).
Salton, G. (1988). *Automatic Text Processing*. Addison-Wesley Publishing Company.
Salton, G., & Buckley, C. (1996). Term weighting approaches in automatic text retrieval. *Technical report TR87-881, Department of Computer Science, Cornell University, 1987. Information Processing and Management, 32*(4), 431–443.
Subasic, P., & Huettner, A. (2001). Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems* [Special issue].
Tao, Y. H., Hong, T. P., & Su, Y. M. (2008). Web usage mining with intentional browsing data. *Expert Systems with Applications., 34*(3), 1893–1904.
Turban, E., & Aronson, J. E. (2001). *Decision support systems and intelligent systems* (6th ed.). Hong Kong: Prentice Internacional Hall.
Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*. Wiley Publishing.
Webometrics. (2009). http://www.webometrics.info/index.html.
Wik, P., & Hjalmarsson, A. (2009). Embodied conversational agents in computer assisted language learning. *Speech Communication, 51*(10), 1024–1037.
Zadeh, L. A. (1994). Fuzzy logic, neural networks and soft computing. *Communications of the ACM, 3*(3), 77–84.
Zhai, J, Wang, Q., & Lv, M. (2008). Application of fuzzy ontology framework to information retrieval for SCM. In *Proceedings of ISIP08, International symposiums on information processing* (pp. 173–177).
Zhang, R., & Zhang, Z. (2003). Addressing CBIR efficiency, effectiveness, and retrieval subjectivity simultaneously. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, MIR*. New York, NY, USA: ACM Press (pp. 71–78).
Zhao, Y., & Karypis, G. (2002). Improving precategorized collection retrieval by using supervised term weighting schemes. In *Proceedings of the international conference on information technology: coding and computing* (pp. 16–21).

Applied Artificial Intelligence: An International Journal

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uaai20

SABIO: SOFT AGENT FOR EXTENDED INFORMATION RETRIEVAL

Ariel Gómez [a] , Carlos León [a] , Jorge Ropero [a] , Alejandro Carrasco [a] & Joaquín Luque [a]

[a] Departamento de Tecnología Electrónica, Universidad de Sevilla, Sevilla, Spain

Version of record first published: 04 Apr 2013.

To cite this article: Ariel Gómez , Carlos León , Jorge Ropero , Alejandro Carrasco & Joaquín Luque (2013): SABIO: SOFT AGENT FOR EXTENDED INFORMATION RETRIEVAL, Applied Artificial Intelligence: An International Journal, 27:4, 249-277

To link to this article: http://dx.doi.org/10.1080/08839514.2013.774204

Applied Artificial Intelligence, 27:249–277, 2013

ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839514.2013.774204

Taylor & Francis
Taylor & Francis Group

SABIO: SOFT AGENT FOR EXTENDED INFORMATION RETRIEVAL

Ariel Gómez, Carlos León, Jorge Ropero, Alejandro Carrasco, and Joaquín Luque
Departamento de Tecnología Electrónica, Universidad de Sevilla, Sevilla, Spain

□ *In the current study, an integrated system called SABIO is presented. The current system applies Information Retrieval (IR) techniques developed for collections of textual documents to non-textual corpa. SABIO integrates a fuzzy logic-based procedure for IR. Its search algorithm improves the IR efficiency and decreases the computational burden by using a fuzzy logic-based procedure for IR. This procedure is integrated in a flexible and fault-tolerant, human-reasoning-based search algorithm. The Accumulated Knowledge Set (AKS) of the system is sorted in a hierarchic multilevel tree-structure-like ontology. The objects in the AKS are represented using a novel human-reasoning-based-method. This representation takes into account the occurrence of related terms. The system uses a novel fuzzy logic-based term-weighting (TW) method. The developed fuzzy logic method improves the classical term frequency–inverse document frequency (TF/IDF) method, generally used for TW. The abovementioned system is the core of a wizard for search into the website of the University of Seville, www.us.es, which is currently in testing.*

INTRODUCTION

The World Wide Web and the Internet allow users to access a wealth of information. This fact and the large quantity, and ever-growing, amount of information available make the demand for Information Retrieval (IR) techniques to increase (Aronson Rindflesch, and Browne,1994; Liu et al. 2001). IR research deals mainly with documents. Achieving both high recall and precision in IR is one of its most important aims. IR has been widely used for text classification (Aronson et al. 1994; Liu et al. 2001) introducing approaches such as Vector Space Model (VSM), k-nearest neighbor method (KNN), Bayesian classification model and Support

Note: SABIO - Classified Information Automatic Retrieval System ("Sistema Automatizado de Búsqueda de Información Ordenada" in Spanish). "SABIO" means "wise" in Spanish.

Address correspondence to Ariel Gómez, Departamento de Tecnología Electrónica, Escuela Politécnica Superior de la Universidad de Sevilla. Calle Virgen de África, 7., 41011 Sevilla, Spain. E-mail: ariel@us.es

Vector Machine (SVM; Lu et al. 2002). In another vein, text mining (TM) techniques provide information derived as a result of the text document contents.

Due to the container environment, retrieved objects are textual (document, web pages, etc.). So, document retrieval systems are widely developed and applied for textual-type set search. Mainly, there are two approaches to the query: either the user provides a few keywords, or the user provides a document to use as a model. The second type of queries achieves a good degree of accuracy, but leads to an important computational load. This method is not suitable for large sets of accumulated knowledge. Furthermore, a keyword-based model has less computational burden and a similar structure to the question that a person would make. This feature is significant in systems where the man-machine interface is the natural language.

One of the most extended methods for keyword-based document content identification is the vector space model (VSM; Raghavan and Wong 1986). The method of representation of nontextual objects proposed in the current study is based on the VSM. In VSM, each document is represented by a set of words present in it (keywords). These keywords are chosen with the help of a stop list. The VSM rejects every matching word. Those remaining are called index terms and represent the document in the system. However, not all index terms are equally important for identifying the document they represent. So, it is necessary to add a factor to indicate its importance. This factor is known as the term-weight (TW).

One of the factors habitually used for term weighting in VSM is the so-called term frequency-inverse document frequency (TF-IDF; Lee, Chuang, and Seamons 1997). This scheme uses all the words present in any document representation as a system vocabulary. Term frequency (TF) is the number of occurrences of the index term in the represented document. Inverse document frequency (IDF) is related to the number of occurrences of the same index term in the other documents in the Accumulated Knowledge Set (AKS; Salton and Buckley 1996). Term weight is the product tf*idf. With this method of document representing, the vector length depends on the number of words present in the document. This feature makes it difficult to compare the documents. Length normalization is applied to equalize the number of terms in all the vectors. However, the number of terms of a vector is usually quite large due to the vocabulary size. This feature makes the computational weight increase, and the method becomes impracticable for large corpa. The similarity between the objects is the distance between both the vectorial representations. One of the most used functions is the cosine similarity.

$$\text{similarity}(Q, D) = \frac{\sum_{k=1}^{N} w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^{N} (w_{qk})^2 \cdot \sum_{k=1}^{N} (w_{dk})^2}} \quad (1)$$

Here, Q is the query representation, D is the document representation, N is the number of index terms available in both representations, wqk is the weight coefficient associated to the k-th index term of the query, and wdk is the weight coefficient associated to the k-th index term of the document.

It should be noted that in the VSM method (and others) objects are represented by parts of themselves, in other words, the words in the document. The main objective of the current study is to develop an information retrieval system that can manage information for any kind of knowledge (objects, experience, legislation, professional execution best practices, etc.) and not just in the textual form. In many cases, the representation of the object cannot be made up of parts of the object itself. Human knowledge is not achieved by incorporating parts of known reality. Humans translate the impulses that they perceive through senses. These are encoded in the protein chains that are stored in the brain (Kandel 2006; Hayashi and Yoshida 2004). All the visitors to the Giralda in Seville stored the data of the experience in their memories: architectural form, size, colors, history, location, and so forth. However, none of the physical components of the monument (bricks, marble piece, tile, plaster, or others) was added to the knowledge base of people who visited. The visitors created a representation of the monument that is stored in their memories. In the same way, in the proposed system, the representations of the real-world objects are built by attributes that are not a part of the objects themselves. Among the several possible ways of representing such objects, the system chooses natural language (NL), because the user query is made that way.

The retrieval effectiveness of an IR method is given by two factors: first, objects related with the query must be retrieved, and second, nonrelated objects must be rejected. The recall parameter is defined as an estimator of the first factor (Ruiz and Srinivasan 1998).

$$\text{recall} = \frac{\text{related retrieved objects}}{\text{total related objects in AKS}} \quad (2)$$

The precision parameter is defined as an estimator of the second factor.

$$\text{precision} = \frac{\text{related retrieved objects}}{\text{total retrieved objects}} \quad (3)$$

In the VSM approach and other IR methods, the query, or the document used as a model, is compared with every document in the

collection. In the proposed fuzzy IR method, the query is compared with only a few objects of the collection that represent the system knowledge.

To do this, the objects belonging to the AKS are grouped in a hierarchical tree structure like ontology. This proposed structure has multiple levels so that each set belonging to a level contains several sets of the lower level. Figure 1 shows the proposed structure. Hierarchical classification of AKS is detailed later in this article.

With the proposed four-level structure, it is simple to identify every object in the AKS by successive approximations without having to analyze all the objects of the knowledge. Figure 2 shows the presented system procedure for recovering the information.

Another feature in the procedures for IR and TW presented in the current study is that it takes into account the relationship between the terms (Gómez et al. 2008). In other IR methods, terms are managed independently from each other. This fact causes the loss of the information given by the compound terms. The fact that the representation of the document should correspond to the meaning must not be forgotten. Some authors include a procedure to take account of the syntax of the sentences in the methods (Chow et al. 2009; Song et al. 2008), others include concept networks to represent the knowledge base (Horng Chen, Chang, and Lee 2003; 2005). SABIO pays attention to this information during the process of rendering objects.

The outcome of the previous IR processes was documents, however, the goal of the TM techniques is to provide new information derived as a result of the contents of the text documents (Ben-Dov and Feldman 2010). This way, SABIO integrates the TM techniques with IR, as it finds new information—objects—derived from text-normalized objects.

So, the current system applies the IR techniques developed for collections of textual documents to nontextual corpa. The current study develops a novel human reasoning-based method to represent

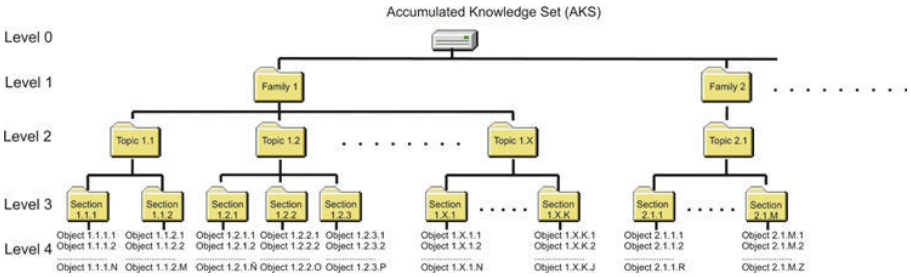


FIGURE 1 Hierarchical tree structure of the AKS. (Color figure available online.)

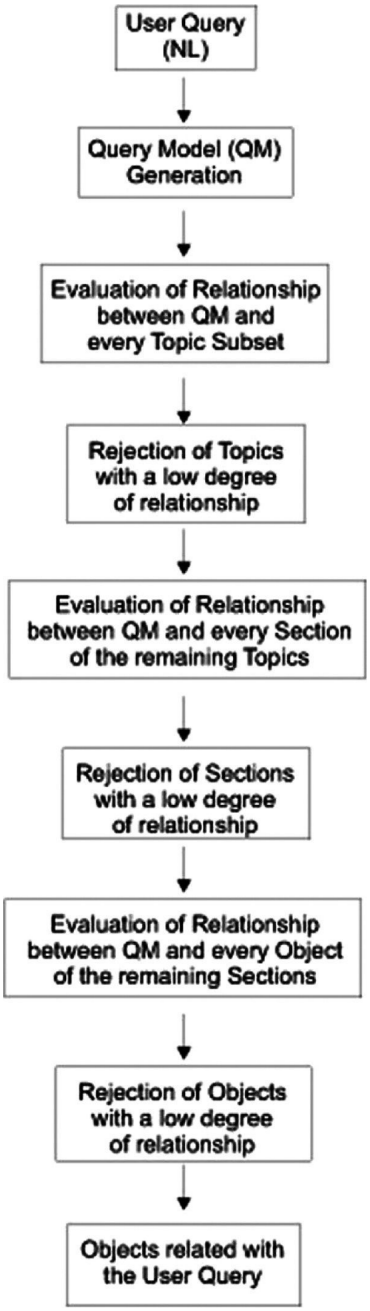


FIGURE 2 System procedure for recovering the information.

objects, taking into account the occurrence of related terms; proposes a fuzzy logic-based term-weighting method; structures the accumulated knowledge on several levels to improve the searches (Bathia and Deogun 1998) and decrease the computational burden; develops a fuzzy logic-based procedure to establish the similarity between a query and an object; and finally, proposes a flexible and fault-tolerant human reasoning-based search algorithm.

The next sections of the article are organized as follows. An accumulated knowledge objects normalization algorithm is introduced in "Accumulated Knowledge Objects Normalization Algorithm." Hierarchical classification of the AKS is described in "Hierarchical Classification of the AKS." The initial configuration of the fuzzy logic-based engine for retrieving the degree of certainty of relationships is explained in "Fuzzy Logic-Based Engine for Relationship Certainty Retrieval: Initial Configuration." The proposed information retrieval algorithm is detailed in "Information Retrieval Algorithm." Level weighting assignation procedure is detailed in "Level Term Weighting." The realized tests, test-derivates system modifications, and model validation are detailed in "Tests, Modifications, and Model Validation." The article ends with conclusions and future work proposals in "Conclusions."

ACCUMULATED KNOWLEDGE OBJECTS NORMALIZATION ALGORITHM

In the above-mentioned IR methods, the objects of the AKS are usually documents. Their representations are built with parts from the objects themselves, in other words, the words contained in them. In SABIO, the AKS objects are not necessarily text-type. Thus, the existing object representation methods are not directly applicable. A general method should be proposed.

Just as the human brain transforms the received information by the senses and stores it permanently in the hippocampus and other structures (Kandel 2006; Hayashi and Yoshida 2004; Sato and Yamaguchi 2010) by using its own cells and proteins, SABIO builds the object representation using parts of the system itself. The bricks used by the system are the terms belonging to its vocabulary. The object representation is not complete without a term-weighting coefficient related to the importance of every word present into the object representation.

So, as object representation, the system uses a set of tuples [a,b], where "a" is a word, and "b" is a related term-weighting coefficient. This transformation procedure is called object normalization. The normalized object representations are stored in a database for future retrieval.

Selection of Index Terms

In a general case of nontextual object, the process of choosing the NL terms to build its description cannot be made independently by direct analysis of the parts of the object itself. In this case, the choice of "a" terms to represent the object within the system is determined as follows.

The person who describes the objects in the set of knowledge is usually called the Knowledge Engineer (KE). The KE builds questions, which answer in NL to describe the object. This kind of sentence is named as a standard question. Another way of building a standard question is to just describe the object. The object representation will be built from a few standard questions.

From this set of standard questions, the KE must extract a few words, rejecting all the words that do not have a real relationship with the object. It should be noted that this fact excludes not only the stop words that were defined earlier, but also more words. A word can be very significant for the description of one object and nonrelevant for another one. This word is kept in the first case and is rejected in the second. The selected set of words that appears in any of the standard questions describing an object is called a set of index terms. Each of these words constitutes the "a" elements of the tuple array that represents the object to the system. In Figure 3 an index-terms selection for object normalization is described.

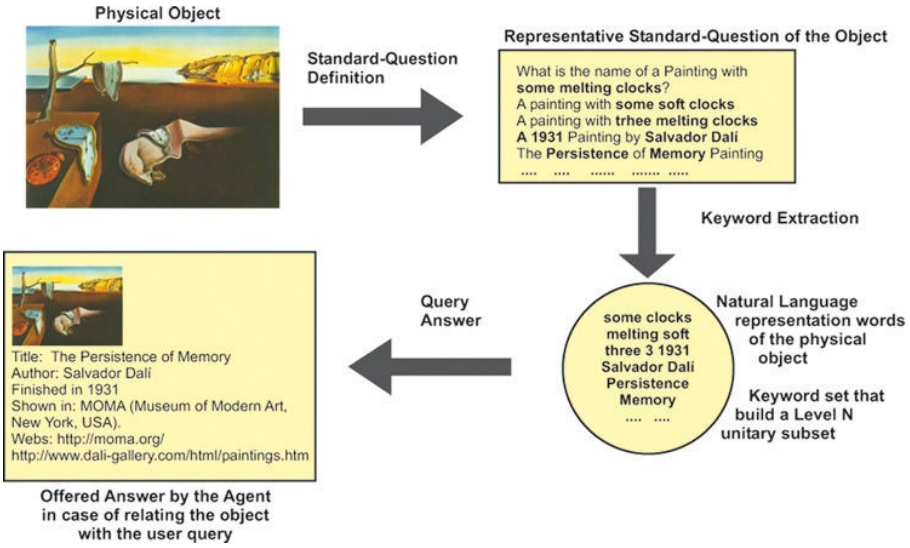


FIGURE 3 Normalization process of the objects in AKS. (Color figure available online.)

It should be noted that using the complete set of index terms for the IR does not make sense, as there is no request containing all the defined index terms. It is obvious that not all the selected index terms have the same relevance in the description of the object. So it is necessary to assign a term weight "b" to each index term "a" to describe this feature (relationship). All the index term sets build the vocabulary of the system. TW normalization is not possible because the representation of the object is not unique.

Term-Weight "b" Purpose

The strength of the relationship between a keyword "a" and the object is expressed by a coefficient. In the current system, this coefficient can take values between 0 and 1. Assigning a 0.00 value means no index-term relationship with the object, whereas assigning a value of 1.00 indicates the highest possible degree of relationship. This concept is related with the TW methods previously described as VSM. "Level Term Weighting" details the algorithm for calculating the "b" component of the tuple. Thus, each object of the AKS is represented by a set of tuples [a,b], where "a" is an index term, and "b" is the value representative of the degree of affinity between the term "a" and the object.

Normalization of the Query by the System

As mentioned previously, the SABIO Human Machine Interface (HMI) is natural language. So, information retrieval is made by a user query in NL. This query will not match exactly with one of the system-defined standard questions to extract object representations. Thus, it must be processed by the system. It is therefore necessary to model the received query but not to classify it, as in Chali (2009), because the answer isn't a document. This modeling process is called query normalization. The representation of the query is the set of words present in the query that match any of those belonging to the vocabulary of the system. SABIO considers only the words that make sense to wake up its memory. It may be noticed that the index terms in a query do not have any associated "b" coefficients.

HIERARCHICAL CLASSIFICATION OF THE AKS

Once all of the objects in the AKS are normalized, there is a whole bag of tuples. Each of them represents an object. When the system receives a query, it must establish the relationship with every object in its AKS. If the system behaved as the current IR methods do, it should make an estimate for every object of AKS. However, if the objects were previously

grouped (by some suitable criterion), the system could determine the affinity of the query with some of the elements of each group, by a single estimate. This argument has two flaws. If small groups are formed, then the procedure is not effective, and if groups are big, the precision of the IR is poor. However, this method should be useful to exclude many objects not related to the received query by a single estimate. This feature causes the rejection of a significant number of objects, reducing the computational burden and processing time for IR. The objects belonging to the subsets not previously rejected could be treated as in the previous case. For this purpose, all objects in a subset should be grouped into smaller subsets defining a second level of grouping. Every set in the second subdivision contains fewer objects than those of the previous level. So fewer objects will be rejected if no relationship with the query is established, but the precision improves. If the last level of aggregation contains singletons, each set corresponds to a single object and the recursive application of this method identifies the objects in the AKS by successive approximations.

It is necessary to find the suitable number of levels of grouping objects so that the identification process provides advantages over the existing ones. It is also necessary to define a clustering approach. Another aspect to determine is the representation format of every group of objects.

Suitable Number of Hierarchic Levels

SABIO proposes to group objects into different sets for every considered level. The common feature for the objects belonging to a set is the existence of the same or similar index terms in their representations in NL. Every set is represented by the union of the NL representations of their component objects. This grouping provides a level of classification with a lower resolution than the previous one.

"Conclusions" validates that grouping the AKS objects in a three-level structure (called topic, section, and object) is enough to improve the efficiency of the subsequent information retrieval about a specific area. The addition of a fourth level (family) of classification of objects should be necessary when the system needs to extract knowledge from significantly different areas.

Representation of the Subsets

In the level structure described, Level N is the highest level (object representation). At this level all the subsets are singletons and the representation is the bag of tuples for every object of the AKS. The next

level (Level N-1) groups, from the previous level, objects that have some common properties. Level N-1 is called the *section level*. Each subset in this section level must have a representation in order to allow the system to determine the relationship with the query. In order to use the same method to establish the relationship to the query, representation of each section subset must have the same structure as that of the objects.

Therefore, the representation of each section consists of an array of tuples [a_s, b_s] (another bag of tuples) The terms "a_s" correspond to union of the the terms "a" present in the representations of the objects belonging to the section. The terms "b_s" establish the relationship of each term "as" with the objects included in the section. The number of tuples in the array is determined by the union of the terms "a" of representations of objects belonging to the section. The term "b_s" associated with each indexterm "a_s" is determined by the values "b" of the representations of objects in which the term "a" appears. Level weighting is detailed in "Level Term Weighting."

The process of grouping several subgroups of AKS in sets containing more objects can be repeated as many times as necessary. At the end of the overall process, the AKS is clustered in different ways at the various levels. Overlapping levels have a pyramid shape. The number of objects in each level is always the same, but the number of subsets grows when closer to the level N. This structure improves the retrieval procedure.

The relationship between an index term "a" with a subset of the AKS is not the same for all the levels because the relevance of the term "a" varies according to the subset in which representation appears. Thus, the "b" term weighing will be different for each level, and the tuples are not the same for every level of the hierarchically structured AKS.

Clustering Criteria

Grouping procedure involves two revisions over the objects of the AKS. The first one is top–down made. Objects are grouped by thematic affinities: topic and section. Once this provisional classification is made, a second bottom–up step is done. Refining criterion is used to put those objects together with the maximum number of common "a" terms in their representations. This criterion is applied only to the grouping of the N-1 level. Most objects are well grouped after the first revision because the common topic usually implies the presence of similar terms. However, at this second step, some objects could be moved from one group to another.

FUZZY LOGIC-BASED ENGINE FOR RELATIONSHIP CERTAINTY RETRIEVAL: INITIAL CONFIGURATION

The aim of the developed system is to answer queries from users without an extensive knowledge of any subject. Therefore, in some cases consultations are expected to be vague and/or nonspecific. Fuzzy logic techniques are suitable for managing this kind of information (Yager and Larsen 1993). The system core is a fuzzy logic engine (FE). FE establishes the degree of relationship between a query and an object or a set of objects in the AKS. The FE receives as input the "b" term of each tuple belonging to the representation of the object to be related; which term "a" matches with any word belonging to the query representation. This process is also applied not only for objects, but for every level of knowledge.

As said in "Term-Weight "b" Purpose," "b" coefficient represents the strength of the relationship between the term "a" and the objects belonging to a certain set. The term "b" is transformed into a linguistic variable that expresses the degree of membership of the term "a" with respect to the subset evaluated. This variable can take three linguistic values: low, medium, and high. Figure 4 shows the aspect of the universe of discourse of this variable.

To answer a user query, the system needs to determine the relationship between the query and one of the objects present in its corresponding AKS. For this task, the system has an FE capable of establishing the degree of certainty for the relationship between the query and one object in the AKS. The determining FE parameters are the number of inputs, the number of outputs, the inference rules, the type of fuzzyficator, and the type of defuzzyficator. The different processes involved in the determination of these parameters are described in the following section.

Fuzzy Logic Rules

To oversimplify, the methodology for determining the degree of relationship between the query and an object in the AKS is based on the values of the chosen "b" terms. Frequently, the higher the "b" terms, the higher the degree of certainty.

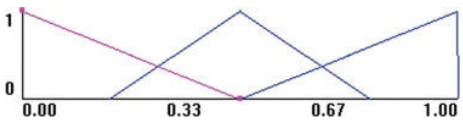


FIGURE 4 Universe of discourse of the input. (Color figure available online.)

The inference rules determine the degree of certainty in the relationship between the query and the object. Obviously, this way of describing the degree of relationship determines fuzzy logic as the better way to solve the proposed task. Thus, the rules that decide the relationship degree between the query and the object will be expressed as: "IF . . . THEN . . . ." sentences. The generally used criterion for defining the rules is as follows: the more inputs with high values, the higher the value of the relationship. The deployment of this approach results in a number of rules depending on the number of FE inputs.

Inputs Number of the Fuzzy Logic Engine

The inputs to the FE are the "b" terms extracted from the query. Ideally, the query should correspond exactly with any of the KE defined standard-questions. Thus, the FE input number is conditioned by the number of words appearing in the representation of objects. This should be sufficient to consider for calculating all of the "b" terms of the tuples of the representation, for every standard question. In most cases, 3 to 5 words are extracted from each defined standard question. Therefore, the use of a three-input engine was initially proposed to assess the certainty of the relationship between an object and the query.

The FE final configuration and the reasoning for it are detailed in "Conclusions."

Fuzzy Logic Engine Output

As described in "Effect of threshold value," the FE has to show a single output: the degree of certainty of the relationship between the query and an object of the AKS. By the nature of the fuzzy rules, the output is a fuzzy value. Thus, its expression is provided by a linguistic expression. The linguistic variable called "certainty of relationship" can take four linguistic values: Low, Medium-Low, Medium-High, and High. Graphically, the shape of this output is shown in the Figure 5.

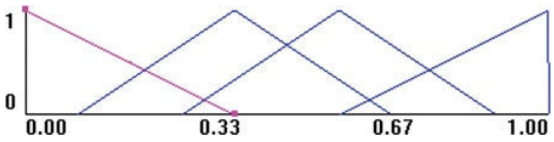


FIGURE 5 Shape of certainty degree of relationship. (Color figure available online.)

Fuzzyfication and Defuzzyfication Methods

Once the input and output numbers and the inference rules are specified, the fuzzyfication and defuzzyfication methods remain to be chosen. Because the output value has to be related to the set of all inputs, and not only to a dominant one, the center of gravity (COG), and mean of maximum (MOM) are considered as the defuzzyfication methods. Initially, the chosen method is COG. For the fuzzyfication method, a generic singleton is elected. "Conclusions" details the tests that lead to the final configuration of the system.

Relationship Certainty Retrieval Algorithm

Determination of the certainty of relationship between the received query and the corresponding subset of the AKS algorithm includes the following steps:

1. Query normalization as described in. "Normalization of the Query by the System" In the end, the query is represented by a word array.
2. Selection of tuples belonging to the evaluated subset representation that term "a" matches with any of the terms of the query representation. If the query representation involves more tuples than FE inputs, those tuples whose "b" terms are lower are rejected. This condition occurs when the number of selected tuples is higher than the FE inputs number.
3. The "b" terms of the selected tuples are the input values to the FE. If any FE input has no associated "b" value, 0.00 is taken as the associated input value. This condition is presented when the number of selected tuples is lower than the FE inputs number.
4. In general, the returned value by the FE is the degree of certainty associated with the relationship of the query with any of the objects contained in the considered subset. Figure 6 shows the described procedure.

INFORMATION RETRIEVAL ALGORITHM

At this point, there is a hierarchically structured AKS. It is structured in several levels, considering three levels for the example given following. There is also a system capable of evaluating the certainty degree of the relationship between a query and a subset of the AKS. There is a need to describe the full information retrieval algorithm used by the system intelligently. The main goal of the algorithm is to be able to discard many of the

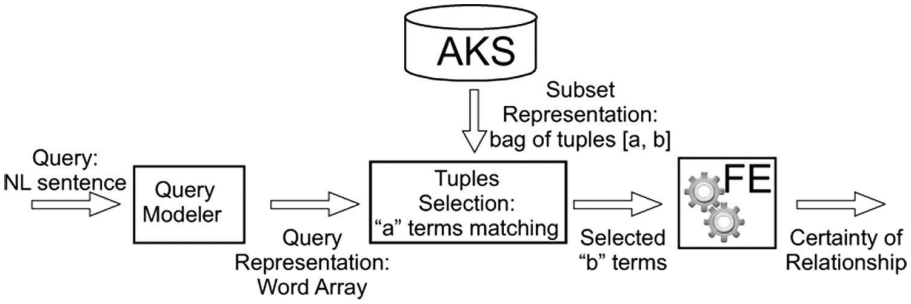


FIGURE 6 Certainty of relationship retrieval algorithm.

objects in the AKS in the early steps. To do this task, the system begins evaluating the certainty degree of the relationship between the query and every first-level subset to see which is the largest by applying the algorithm described in "Relationship Certainty Retrieval Algorithm."

A threshold value is established for every level. The purpose of this threshold is to reject those subsets with a lower certainty of relationship obtained in the previous step. In this manner, many objects are rejected by only one estimation. Thus, the computational efficiency of the algorithm increases. Now, the process is applied again to those subsets that obtained a degree of certainty higher than the threshold, but using the next level of classification. The aim of the process is to approach the query-related objects without evaluating every object of the AKS. This search refines the results using the subsets present in the following levels. Only subsets corresponding to the accepted sets of the previous level are considered. Figure 7 illustrates the procedure, considering a three-level structured AKS.

The first step is to normalize the query. The representation of the query consists of a word array without the associated coefficients instead of a set of

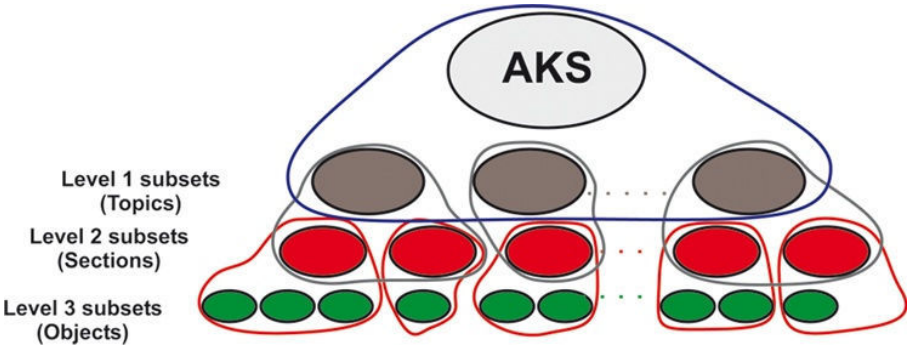


FIGURE 7 Three-level structured AKS. (Color figure available online.)

tuples. As mentioned previously, selected words should be contained in the system vocabulary.

The second step is to determine the degree of certainty that the query is related to any of the objects in first-level subset (Topic). For every subset in this level, the system takes the set of tuples representing the subset for which the relationship is being evaluated. The system selects those tuples whose "a" term matches any word present in the representation of the query.

In the next step, the FE inputs are fed with the associated "b" terms of the previously selected tuples. FE output is the degree of certainty of the relationship between the query and some of the objects in the specific subset. This procedure is applied one by one to every first-level subset. At the end, the system has a certainty value associated with each first-level subset.

The last step for this level is to reject those subsets whose associated certainty value is lower than a predetermined threshold. In Figure 8, only the first and the last subsets are above the threshold. Thus, the remaining subsets are rejected.

The same procedure is applied to each Level-2 subset belonging to those Level-1 subsets for which the certainty value was higher than the threshold. As a result, a new array of certainty values decides which Level-2 subsets are rejected. Only those Level-2 subsets whose associated value is greater than the Level-2 threshold will remain. Note that the threshold for Level-1 does not have to be the same as that of Level-2.

In Figure 9, only the second and the third Level-2 subsets are accepted. Those remaining are rejected.

At this point, only three objects of the AKS would be related with the query. To determine which are finally related, the above procedure is applied one more time to the objects belonging to the remaining subsets. This time, Level-3 subsets are the object representations themselves. All

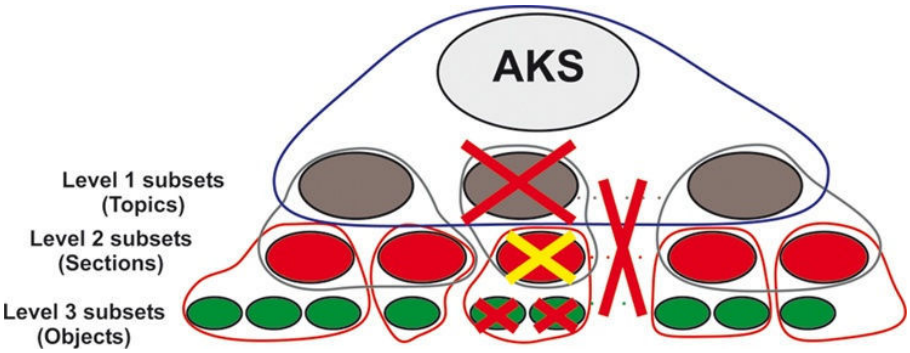


FIGURE 8 Example of first-level evaluation. (Color figure available online.)

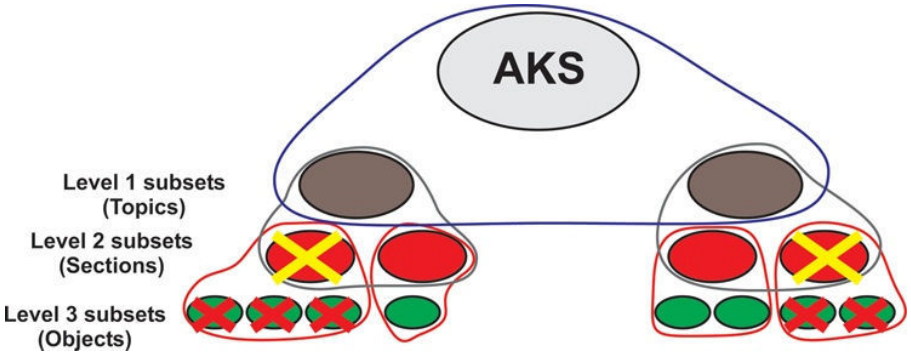


FIGURE 9 Example of second-level evaluation. (Color figure available online.)

those objects whose value of certainty exceeds the Level-3 threshold are identified as related to the query.

Effect of Threshold Value

The threshold value can significantly affect the operation of the system. If a high value is set, only the objects that are strongly related to the query are considered. If the setting is low, fewer objects are rejected, thereby decreasing the efficiency and precision parameters. Thus, it might seem that high values would improve the efficiency of the system. Nevertheless, it should not be forgotten that the HMI is the natural language, and the user is not necessary skilled in the art. If the query is vague or imprecise, the recall parameter would decrease significantly. The test results showed 0.5, 0.55, and 0.65 threshold values for Level 1, Level 2, and Level 3, respectively, which is a configuration that leads to good results when the users have some knowledge about the subject being queried. Fixing the threshold value to 0.5 for all the levels is a more general configuration that also shows good results. In this case, no special requirements are needed from the users.

Another problem related to the thresholds occurs when none of the subsets of a given level has a certainty value higher than the required threshold. In this case, the system response would be: none object related. However, the real problem could be a too-high threshold value, a weak relation, or a missed coefficient in the database.

In order to avoid quitting the procedure in the early stages, the system applies a human reasoning so-called *inkling theory* (Gabora 2000). When a person is asked about something and is obviously reminiscent, a relationship between the question and the memories can be established with great certainty. However, when a person is unable to relate any memories so

strongly, an attempt is made to link weak memories. In fact, the followed process reduces the level of demand to the certainty of the relationship between the question and the related memory. The analogous process followed in the system is to automatically lower the threshold value at the level where none of the subsets takes a value of certainty sufficient to reach the initially set threshold. When none of the subsets in a level take a value of certainty to reach the threshold, SABIO decreases the threshold value on 0.05 steps. This reduction takes place until one of the subsets obtains a value that reaches the new threshold. It should be noted that the other threshold levels remain unchanged, and the modified threshold assumes its original value for new queries. Often, more than one object exceeds the reduced threshold. All of them are accepted and the others are rejected.

LEVEL TERM WEIGHTING

As a result of the procedure described in "Information Retrieval Algorithm," every object representation needs a "b" coefficient for every defined level. So, for each level, a "b" coefficient associated to the "a" terms belonging to each subset must be calculated. This calculation is one of the most important tasks for IR. To solve the problem, many authors have considered VSM and, specifically, have used the TF-IDF method. In this section, a novel alternative fuzzy logic-based method for TW has been proposed.

In the current proposal, not just the statistical parameters are included in the weighting calculation, but the meaning of the term "a" and the possibility of it being part of a compound term is also taken into account. Thus, TW includes the influence of the affinity between the meaning of the index term and the object itself. The proposed TW scheme is a fuzzy logic-based product of the two meaning-based parameters mentioned earlier, plus two other TF-IDF–based statistical parameters.

Therefore, in the proposed weighting method, the assigned value for the coefficient "b" is related to four parameters that can take values between 0.00 and 1.00. The four proposed parameters and their influence are detailed in the next subsection.

Weighting Related Parameters

The first and most significant parameter is the degree to which the term "a" undoubtedly identifies the object without any other term present in the query. The more identification, the higher is the parameter value. This parameter is a new approach for introducing semantic information in the object representation. An expert in the matter should intuitively

evaluate the importance of the "a" terms. This method is simple, but it has the disadvantage of depending exclusively on the KE. It is very subjective and not possible to completely automate the method. This parameter has no correspondence to any previous method in IR.

The parameter value is given by Table 1.

The second parameter depends on the frequency of occurrence of the term "a" in the representations of the other subsets at the same level in the AKS. The higher the frequency, the lower is the parameter value. This parameter is related with the classical VSM concept of IDF but, in the current case, the assigned value is obtained through a table, and not by any of the usual formulae.

For the construction of the table it was considered that 1% of the most-frequently used words present in the vocabulary define the border for the value 0.00. The most-frequently used words should be understood as those belonging to a higher number of other subset representations. This ranking is made for every considered level. For example, if the vocabulary is 1000 words in size, the one that ranks tenth in the number of appearances in the other subsets of a specific level indicates the number of occurrences for which the parameter value is 0.0 for the considered level.

Continuing the example, it is assumed that the tenth word belongs to the representation of thirteen subsets. An "a" frequency of occurrence greater than or equal to 13 leads to a 0.0 value for this second parameter. This parameter is easily computable by the system, so the table will have 13 columns and 2 rows. The number of occurrences is in the first row, while the associated second parameter value is in the second one. The 0.00 to 1.00 range is divided among the thirteen possible values. The values for the considered example are shown in Table 2.

The third parameter depends on the number of object representations belonging to the same subset where the "a" term appears. The more objects in a set an "a" term belongs to, the higher the value for the corresponding parameter value. This parameter is related with the classical VSM concept of TF but, in our case, the assigned value is again obtained through Table 3.

In the same manner, 1% of the most-often used words define the boundary value 1.00. Consider the same example given previously. In the new most-often used word list, the tenth one sets the number of occurrences from which the parameter value is 1.00. Now, the most-often used

TABLE 1 First Weighting Parameter Value

Does this "a" term undoubtedly define the object by itself?	Yes	Rather	No
1st parameter value	1.0	0.5	0.0

TABLE 2 Second Weighting Parameter Value

Subsets representation to which "a" belongs	0	1	2	3	4	5	6	7	8	9	10	11	12	≥13
2nd Parameter Value	1.00	0.90	0.80	0.70	0.64	0.59	0.53	0.47	0.41	0.36	0.30	0.20	0.10	0.00

words should be intended as those belonging to a higher number of object representations, in the same subset, for the considered level. If the tenth "a" term belongs to five object representations, any "a" term representing six or more objects in a subset takes a value of 1.00 for this parameter.

This parameter is easily computable by the system. Note that this parameter is senseless at the level of the object because all of the subsets are singletons.

The fourth parameter is related to the possibility that the term "a" belongs to a compound term, (i.e., web, mail, and web mail). This parameter increases the semantics precision of the representative ghost of the object. Four cases are considered and the corresponding parameter value is shown in Table 4.

A different approach of including the related terms effect can be found in Chow, Zhang, and Rahman (2009). The system related in the current study is not capable of evaluating this parameter by itself because of the nature of the representation of the objects. Because the relationship between the value of the four parameters involved in the TW task and the final value of "b" term weight coefficient is difficult to express numerically, it seems more appropriate to use a fuzzy reasoning. So, the FE described in "Accumulated Knowledge Objects Normalization Algorithm" is adapted to determine the "b" term value using the four described parameter values as inputs.

Fuzzy Weighting Rules

Now, the FE inputs are the four weighting-parameter values, and the possible input values are High (H), Medium (M), and Low (L). The new output is the "b" value. The possible output values of "b" are High (H), Medium-High (MH), Medium-Low (ML), and Low (L).

TABLE 3 Third Weighting Parameter Value

Object representation to which "a" belongs	1	2	3	4	5	≥6
3rd Parameter Value	0.00	0.30	0.45	0.60	0.70	1.00

TABLE 4 Fourth Weighting Parameter Value

Number of tied "a" terms to the considered one	0	1	2	>2
4th Parameter value	1.00	0.70	0.30	0.00

Another set of rules is defined for the new purpose. Table 5 summarizes the rules.

A system prototype was created to test the performance of the weighting method proposed. This prototype was implemented using Borland C++Builder.

Reduction of Human Dependence

The first and fourth parameters described in "Weighting Related Parameters" require the intervention of a person, preferably a KE, to assign a specific value to them. To avoid this dependence as much as possible and minimize the qualification of the person in charge, the specification requirement is reduced to answering two questions in NL. The first question is: "Does this "a" term undoubtedly define the object by itself?" The response has only three possible values: Yes, Rather, or No. Those values correspond to inputs High, Medium, or Low, respectively. Table 1 shows the possible numerical values for this parameter.

The second question is: "Is this "a" term tied to another one?" The response has only four possible values: "to none," "to another one," "to another two," or "to more than 2." Table 4 shows the possible numerical values for this parameter. These questions are easy enough for anyone introducing knowledge into the system to be able to answer without any special requirements. The goal is to answer these two questions when the object is added to the AKS system as an integral part of the process of adding new objects. With these two parameters, the system has all the data to

TABLE 5 TW Rules

Rule n°	Rule definition	Output
R1	IF P2 = H, AND P3 $\neq$ L	At least MH
R2	IF P2 = M, AND P3 = H	At least MH
R3	IF P2 = L, AND P3 = L	Depends on other questions
R4	If P2 = H, AND P3 = H	Depends on other questions
R5	IF P1 = H	At least MH
R6	IF P4 = L	Descends a level
R7	IF P4 = M	If the output is ML, it descends to L
R8	IF (R1 and R2) OR (R1 and R5) OR (R2 and R5)	H
R9	Any other case	ML

determine by itself the last level of "b" term values. For higher levels of the AKS, the value taken by the "b" term is the average of the lower levels.

Additionally, there are two important advantages for the new method. On the one hand, TW is close to being automatic, whereas on the other hand, the level of required expertise is much lower. This is because there is no need for an operator to know much about the way FE works, but only to know how many times a keyword appears in every set and the answer to two simple questions: "Does a keyword undoubtedly define an object by itself?" and "Is a keyword tied to another one?"

In "Term Weighting Test," a test comparing the TF-IDF method and the fuzzy logic-based one was performed.

TESTS, MODIFICATIONS, AND MODEL VALIDATION

A desktop application was created to test the performance of the whole proposed method. This prototype was implemented using Borland C++Builder. Figures 10 and 11 shows its main windows.

The prototype can generate a report detailing the reasoning followed by the system, as shown in Figure 12. This feature has proven to be very helpful in debugging the faults found by determining the failure causes and correcting them.

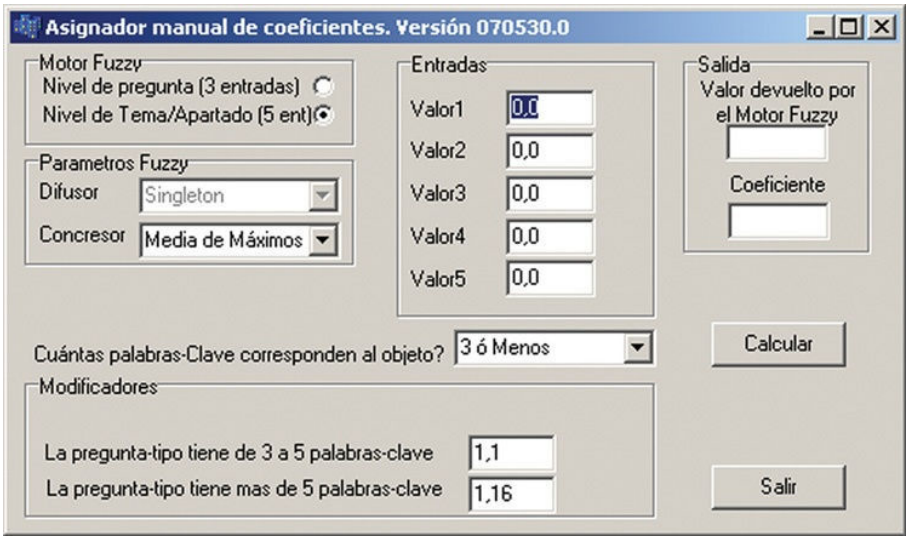


FIGURE 10 Main window application for testing the proposed TW method. (Color figure available online.)

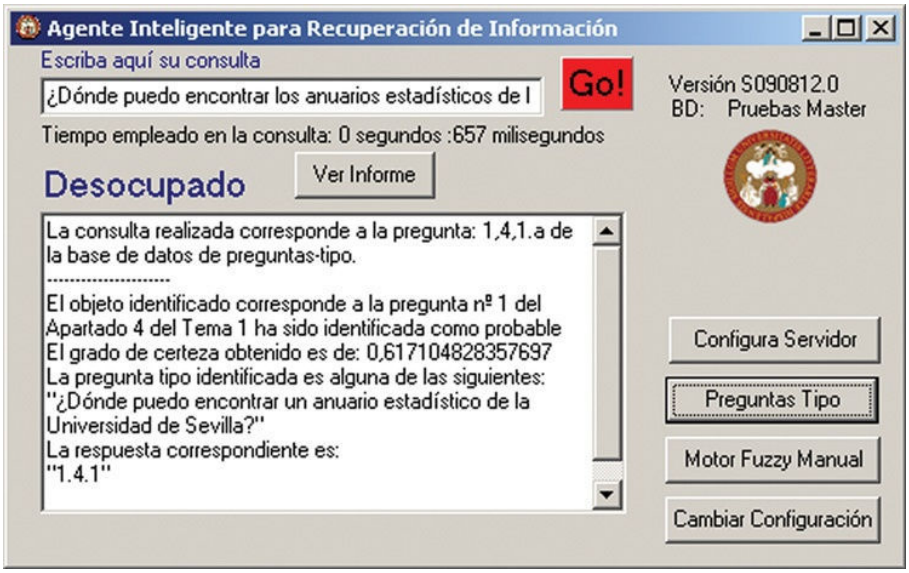


FIGURE 11 Main window of application for testing whole proposed method. (Color figure available online.)

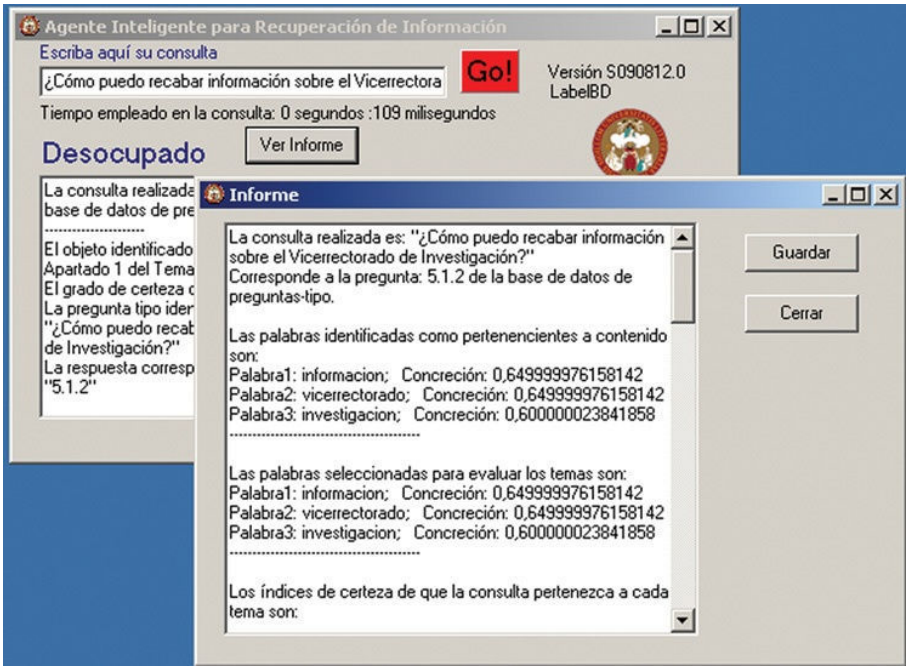


FIGURE 12 Report of the reasoning applied. (Color figure available online.)

For the initial tests, a FAQ set was taken as an AKS system. Every question was treated as an object. All of these objects were normalized, and many standard questions were added to take into account other same-meaning expressions. For this initial test, the TW assignment was manually made, and all the coefficients were kept in a Microsoft Access data base.

The AKS system was built in a three-level structure: Topic, Section, and Object. The system consists of 12 Topics. Every topic is divided into a number between 3 and 12 Sections, and each Section contains between one and eight Objects.

The used validation test was called a "self-test." It consists of feeding the system with its own standard questions. Although potential users probably would not use the exact same standard questions, the aim of this test is obvious: the system must identify the normalized representation of each object in the AKS. Thus, the system's standard questions will be used as a query. Moreover, the certainty of the recognition should have a high value, over 0.7.

The prototype is provided with an interface to change the relevant parameters of its configuration. Thus, when the partial test result was admissible, an entire self-test was made.

The position in which the correct answer appeared among the retrieved answers is considered in order to compare the self-test results. The results are grouped into five categories. If the test result belongs to one of the first four (a, b, c, d), it is considered satisfactory. On the contrary, if it belongs to the last (e), the result is considered unsatisfactory. The meaning of each category is shown in Table 6.

For the self test, the configuration of the FE was as follows: input number–3; output number–1; fuzzyficator–singleton; defuzzyficator–COG; thresholds–0.5, 0.5, 0.5 fixed. The obtained results for the first test are shown in Table 7. The test results show a good performance of the method when the object is represented from two to four tuples. For the objects represented by more tuples, the system displays a tendency to consider them related, even when they are not related or are just nearly related.

TABLE 6 Possible Test Results Grouped by Categories

Category	Meaning
a	The correct answer is retrieved as the only answer or it is the one that has a higher degree of certainty among the answers retrieved by the system.
b	The correct answer is retrieved among the two with a higher degree of certainty—excluding the previous case.
c	The correct answer is retrieved among the three with a higher degree of certainty—excluding the previous case.
d	The correct answer is retrieved, but not among the three with a higher degree of certainty.
e	The correct answer is not retrieved by the system.

TABLE 7 Categorized Results of Self-Tests

Category	a (%)	b (%)	c (%)	d (%)	e (%)
1st test	43.51	24.22	8.59	11.72	10.16
2nd test	54.89	12.03	3.01	0.75	29.32
3rd test	69.93	14.29	3.00	0.75	12.03
4th test	77.44	15.79	4.51	0.75	1.51

The test was repeated using a five-input FE and the same settings for the rest of the parameters. The obtained results for this second test are also shown in Table 7. A positive observed effect is that the "a" category improved their performance, which means that the precision increases. On the other hand, the "e" category increases their matching, thereby indicating that recall gets worse. Analyzing the reasoning reports of the system results belonging to the category "e," it becomes clear that the failure appears because none of the subsets exceed the required threshold in some stage of the procedure. This fact encourages changing the algorithm for determining the relationship. According to this, if no subset has a relationship certainty above the required threshold, the threshold value is decreased in 0.05 steps until any subset exceeds it.

A third test was done using the adaptive threshold algorithm and the same configuration as above. The test results are also shown in Table 7. Many more answers were observed in the "a" category, whereas many fewer answers were observed in the "e" category. This means that the introduced changes to the algorithm improve recall and precision. Analyzing the reasoning report provided by the system, applied to the objects of the category "e," it is clear that the procedure assigns a lower value for the certainty of relationship in a query when the object is represented by three or fewer tuples. This finding encourages a new change in the IR procedure. According to this fact, if the object representation has three or fewer tuples, a three-input FE is used to determine the degree of certainty of relationship in a query. Otherwise, a five-input FE is used. This new modification is related in the same manner with the VSM normalization concept. However, the proposed scheme is significantly simpler and does not require a recalculation of all the coefficients in the case of a change in the vocabulary of the system. In both cases the adaptive thresholds algorithm is applied.

A fourth test—another autotest— was made using the last procedure modification. The configuration of the FE was as follows: number of inputs–either three or five, depending on the index terms extracted from a query; number of outputs–1; fuzzyficator–singleton; defuzzyficator–COG; thresholds–0.5, 0.5, 0.5 adjustable. The obtained results are shown in Table 7.

The obtained results with this last configuration show significant improvement over any one of the earlier configurations. An increase in both recall (98.49%) and precision (77.44%) was observed. Therefore, it is considered that the results validate the algorithm for determining the degree of certainty of the relationship with the query, and the proposed IR procedure.

Fuzzy Logic Engine Optimization

Once the IR process is set and validated, it is desirable to optimize the FE core. A battery of tests is specified to determine the best fuzzyficator, defuzzyficator, and the most suitable type of universe for the inputs and the output. Table 8 shows the settings of the six proposed self-tests.

The autotests results are shown in Table 9.

The analysis of these results shows the following:

- The couple triangle fuzzyficator and COG defuzzyficator obtain more "e" category results regardless of the type of universe of the inputs and the output.
- The couple singleton and COG obtain more "a" category results in the curved universe for the inputs and the output.
- The couple singleton and COG obtain more "a" + "b" + "c" categories results in the orthogonal universe for the inputs and the output.
- The couple singleton and COG obtain fewer "d" + "e" categories results in the orthogonal universe for the inputs and the output.

Thus, it was concluded that the optimal configuration uses a singleton fuzzyficator, a COG defuzzyficator, a straight input universe, and a straight output universe.

Term Weighting Test

To validate the usefulness of the proposed fuzzy logic-based weighting method, a comparative test between the classical TF-IDF method and the proposed one was suggested. Some of these results were presented in

TABLE 8 Proposed Self-Test to Improve the FE Core Performance

Test n°	Fuzzyficator	Defuzzyficator	Input universe	Output universe
1	Singleton	COG	Straight	Straight
2	Triangle	COG	Straight	Straight
3	Singleton	MOM	Straight	Straight
4	Singleton	COG	Curved	Curved
5	Triangle	COG	Curved	Curved
6	Singleton	MOM	Curved	Curved

TABLE 9 Categorized Results of Improvement of the FE Core Performance Self-Tests

Category test n°	a	b	c	d	e
1	77.44	15.79	4.51	0.75	1.51
2	69.17	18.05	3.76	5.26	3.67
3	68.42	15.04	6.77	7.16	2.26
4	75.94	15.79	4.51	1.50	1.50
5	84.21	8.21	1.50	2.26	3.76
6	65.41	18.78	6.02	8.27	1.50

advance in Ropero et al. (2009). A new AKS was built using the objects belonging to the web portal of the University of Seville. This web portal has 50,000 daily visits, which qualifies it into the 10% most visited university portals, and it is ranked 190 among more than 20,000 Universities in the Webometrics rankings for Universities' web impact (Webometrics 2011). Becasue the information in the university web portal is abundant, 253 objects grouped in 12 topics were defined. All these groups were made up of a variable number of sections and objects. 2107 standard questions surged from these 253 objects. However, slightly more than half of these questions were eliminated for these tests because of being very similar to others. Eventually, the tests consisted of 914 possible user queries.

The formula to obtain the TW coefficient using the TF-IDF product has been modified and improved by many authors to achieve better results in IR and IE. Eventually, the chosen formula for the current tests was the one proposed by Liu et al. (2001)

$$W_{ik} = \frac{tf_{ik} \times \log(N/n_k + 0.01)}{\sqrt{\sum_{k=1}^{m} \left(tf_{ik} \times \log(N/n_k + 0.01)\right)^2}}, \tag{4}$$

where tf_{ik} is the ith term frequency of occurrence in the kth subset— Topic/Section/Object—and nk is the number of subsets to which the term Tk is assigned in a collection of N objects. Consequently, it has been taken into account that a term might be present in other sets of the collection.

It was suggested to present between 1 and 5 answers, depending on the number of related Objects. The results of the consultation were sorted in

TABLE 10 Categorized Results of TF-IDF vs. SABIO Self-Test

Category	a	b	c	d	e
TF-IDF	466 (50.98%)	223 (24.40%)	53 (5.80%)	79 (10.18%)	93 (10.18%)
SABIO	710 (77.68%)	108 (11.82%)	27 (2.95%)	28 (3.06%)	41 (4.49%)

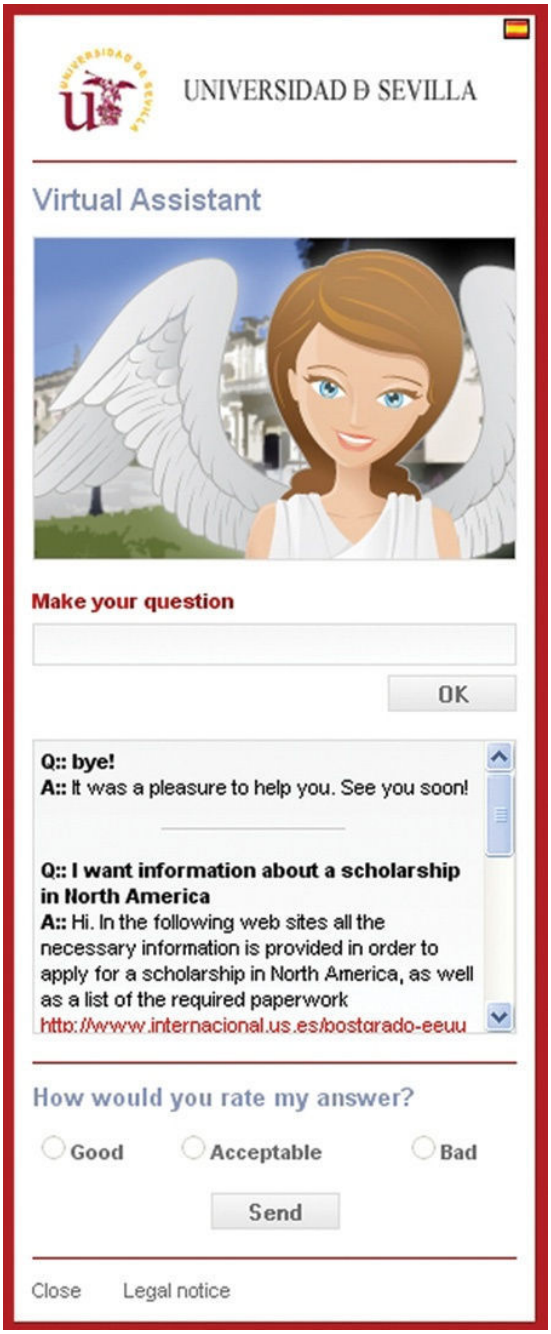


FIGURE 13 Wizard for navigation on the website of the University of Seville. (Color figure available online.)

the same five categories as those in Table 6, titled "Possible Test Results Grouped by Categories." The ideal situation comes when the desired Object is retrieved as "a," though "b" and "c" would be reasonably acceptable. The obtained results are shown in Table 10.

Although the obtained results with the TF-IDF method are quite reasonable, 81.18% of the objects being retrieved among the first five options and more as "a" category, the fuzzy logic-based method turns out to be clearly better, with 92.45% of the desired objects retrieved and more than three-quarters as the first option.

CONCLUSIONS

The current study presents an Information Retrieval system that is able to manage information relating to any kind of knowledge (objects, experience, legislation, professional execution best practices, etc.), and not only to textual knowledge. The human-system interface is natural language.

The hierarchical structure for information classification and storage proposal, in conjunction with the retrieval procedure of the objects related to the query, leads to a lower required computational load, unlike most of the existing procedures.

A novel fuzzy logic-based algorithm for determining the certainty of the relationship between a query and its corresponding subset of the AKS is developed.

The article also presents a novel fuzzy logic-based term weighting algorithm. This novel TW algorithm is easy to use and requires no specialized knowledge. Tests show that this novel algorithm improves the performance when compared to the widely spread classical TF-IDF.

The system described in the current study is being implemented in the development of a Wizard of contents for the website of the University of Seville. At the present time, the Wizard is in a state of internal testing and will shortly be put into production. Figure 13 shows the appearance of the prototype of the application. In the same manner, the presented system can be used to manage information relating to any matter if queries utilize natural language.

The system presented can also be integrated in a multiagent system (MAS) environment in order to manage more complex knowledge. To achieve this goal, complex knowledge has to be able to be split into several simple components parts. Once the complex information is split into several simple faces, the MAS dedicates a soft agent to manage every simple aspect of the whole knowledge. The MAS system should be provided with a special agent to manage and split the received user query. Other special agents in charge of composing the received simple information should also

exist. A more complex answer must be built from the received information from the several existing soft agents.

REFERENCES

Aronson, A. R., T. C. Rindflesch, and A. C. Browne. 1994. Exploiting a large thesaurus for information retrieval. In *Proceedings of RIAO*, New York, 197–216.

Bathia, S. K., and J. S. Deogun. 1998. Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 28 (3): 427–436.

Ben-Dov, M., and R. Feldman. 2010. Text mining and information extraction. *Data mining and knowledge discovery handbook* doi: 10.1007/978-0-387-09823-4_42.

Chali, Y. 2009. Question answering using question classification and document tagging. *Journal of Applied Artificial Intelligence* 23 (6): 500–521.

Chow, T. W. S., H. Zhang, and M. K. M. Rahman. 2009. A new document representation using term frequency and vectorized graph connectionists with application to document retrieval. *Expert Systems with Applications* 36:12,023–12,035.

Gabora, L. 2000. Toward a theory of creative inklings. In *Art, technology, and consciousness*, ed. R. Ascott, 159–164. Oxford: Intellect Press.

Gómez, A., J. Ropero, C. León, and A. Carrasco. 2008. A novel term weighting scheme for a fuzzy logic based intelligent web agent. In ICEIS 2008–*Proceedings of the 10th international conference on enterprise information systems*, Barcelona: AIDSS, 496–499.

Hayashi, H., and Y. Motoharu. 2004. A memory model based on dynamical behavior of the hippocampus. In *Lecture notes in computer science* 3213/2004:967–973, doi: 10.1007/978-3-540-30132-5_130.

Horng, Y. J., S. M. Chen, Y. C. Chang, C. H. Lee. 2003. Automatically constructing multirelationship fuzzy concepts networks for document retrieval. *Journal of Applied Artificial Intelligence* 17 (4): 303–328.

Horng, Y. J., S. M. Chen, Y. C. Chang, C. H. Lee. 2005. A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE Transactions on Fuzzy Systems* 139 (2): 216–228.

Kandel, E. R. 2006. *In search of memory: The emergence of a new science of mind*. New York, NY: W.W. Norton.

Lee, D. L., H. Chuang, and K. Seamons. 1997. Document ranking and the vector-space model. *IEEE Software* 14 (2): 67–75.

Liu, S., M. Dong, H. Zhang, R. Li, and Z. Shi. 2001. An approach of multi-hierarchy text classification. In *Proceedings of the international conferences on info-tech and info-net* 3:95–100. Beijing.

Lu, M., K. Hu, Y. Wu, Y. Lu, and L. Zhou. 2002. SECTCS: Towards improving VSM and naive Bayesian classifier. *IEEE international conference on systems, man and cybernetics*, Hammamet, Tunisia, 5:5.

Raghavan, V. V., and S. K. Wong. 1986. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science* 37 (5): 279–87.

Ropero, J., A. Gómez, C. León, and A. Carrasco. 2009. Term weighting: Novel fuzzy logic based method vs. classical tf-idf method for web information extraction. In *ICEIS 2009–Proceedings of the 11th international conference on enterprise information systems, AIDSS*, Milan, Italy, 130–137.

Ruiz, M., and P. Srinivasan. 1998. Automatic text categorization using neural networks. In *Advances in classification research 8: Proceedings of the 8th ASIS SIG/CR Classification Research Workshop*, ed. E. Efthimiadis, 59–72. Medford, NJ: Information Today.

Salton, G. and C. Buckley. 1996. Term weighting approaches in automatic text retrieval. *Information Processing and Management* 32 (4): 431–443.

Sato, N., Yamaguchi, Y. 2010. Simulation of human episodic memory by using a computational model of the Hippocampus. *Advances in Artificial Intelligence* 2010, Article ID 392868, 10 pages, doi: 10.1155/2010/392868.

Song, Y. I., K. S. Han, S. B. Kim, S. O. Park, and H. C. Rim. 2008. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems* 31:265–286, doi 10.1007/s10844-007-0045-0.

Webometrics http://www.webometrics.info/index.html. 2011.

Yager, R., and H. Larsen. 1993. Retrieving information by fuzzification of queries. *Journal of Intelligent Information Systems* 2:421–441.

9

Term Weighting for Information Retrieval Using Fuzzy Logic

Jorge Ropero, Ariel Gómez, Alejandro Carrasco,
Carlos León and Joaquín Luque
Department of Electronic Technology, University of Seville, Spain

1. Introduction

The rising quantity of available information has constituted an enormous advance in our daily life. However, at the same time, some problems emerge as a result from the existing difficulty to distinguish the necessary information among the high quantity of unnecessary data. Information Retrieval has become a capital task for retrieving the useful information. Firstly, it was mainly used for document retrieval, but lately, its use has been generalized for the retrieval of any kind of information, such as the information contained in a database, a web page, or any set of accumulated knowledge. In particular, the so-called Vector Space Model is widely used. Vector Space Model is based on the use of index terms, which represent some pieces of knowledge or Objects. Index terms have associated weights, which represent the importance of them in the considered set of knowledge.

It is important that the assignment of weights to every index term - called Term Weighting - is automatic. The so-called TF-IDF method is mainly used for determining the weight of a term (Lee et al., 1997). Term Frequency (TF) is the frequency of occurrence of a term in a document; and Inverse Document Frequency (IDF) varies inversely with the number of documents to which the term is assigned (Salton, 1988). Although TF-IDF method for Term Weighting has worked reasonably well for Information Retrieval and has been a starting point for more recent algorithms, it was never taken into account that some other aspects of index terms may be important for determining term weights apart from TF and IDF: first of all, we should consider the degree of identification of an object if only the considered index term is used. This parameter has a strong influence on the final value of a term weight if the degree of identification is high. The more an index term identifies an object, the higher value for the corresponding term weight; secondly, we should also consider the existance of join terms.

These aspects are especially important when the information is abundant, imprecise, vague and heterogeneous. In this chapter, we define a new Term Weighting model based on Fuzzy Logic. This model tries to replace the traditional Term Weighting method, called TF-IDF. In order to show the efficiency of the new method, the Fuzzy Logic-based method has been tested on the website of the University of Seville. Web pages are usually a perfect example of heterogeneous and disordered information. We demonstrate the improvement introduced by the new method extracting the required information. Besides, it is also possible to extract related information, which may be of interest to the users.

2. Vector Space Model and Term Weighting

In the Vector Space Model, the contents of a document are represented by a multidimensional space vector. Later, the proper classes of the given vector are determined by comparing the distances between vectors. The procedure of the Vector Space Model can be divided into three stages, as seen in Figure 1 (Raghavan & Wong, 1986):

- The first step is document indexing, when most relevant terms are extracted.
- The second stage is based on the introduction of weights associated to index terms in order to improve the retrieval relevant to the user.
- The last stage classifies the document with a certain measure of similarity.

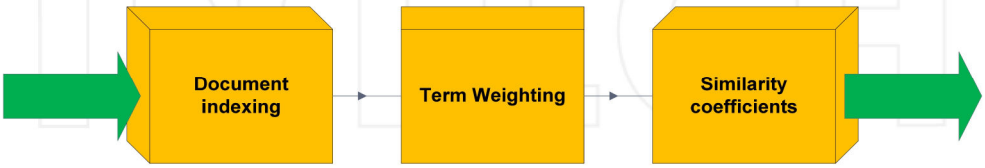


Fig. 1. Vector Space Model procedure

In this chapter, we are focusing in the second stage. It was in the late 50's when the idea of text retrieval came up - a concept that was later extended to general information retrieval -. Text retrieval was founded on an automatic search based on textual content through a series of identifiers. It was Gerard Salton who laid the foundations for linking these identifiers and the texts that they represent during the 70's and the 80's. Salton suggested that every document could be represented by a term vector in the way $D = (t_i, t_j, \ldots, t_p)$, where every t_k identifies a term assigned to a document D. A formal representation of the vector D leads us not to consider only the terms in the vector, but to add a set of weights representing the term weight, it is to say, its importance in the document.

A Term Weighting system should improve efficiency in two main factors, recall and precision. Recall takes into account the fact that the objects relevant to the user should be retrieved. Precision considers the fact that the objects that are not wanted by the user should be rejected. In principle, it is desirable to build a system that rewards both high recall, - retrieving all that is relevant - and high precision - discarding all unwanted objects (Ruiz & Srinisavan, 1998). Recall improves using high-frequency index terms, i.e. terms which occur in many documents of the collection. This way, it is expected to retrieve many documents including such terms, and thus, many of the relevant documents. The precision factor, however, improves when using more specific index terms that are capable of isolating the few relevant articles of the mass of irrelevant. In practice, compromises are utilized; using frequent enough terms to achieve a reasonable level of recall without causing a too low value of precision. The exact definitions of recall and precision are shown in Equations 1 and 2.

$$\text{Recall} = \frac{retrieved\ relevant\ objects}{total\ number\ of\ relevant\ objects}$$

Equation 1. Definition of recall

$$\text{Precision} = \frac{retrieved\ relevant\ objects}{total\ number\ of\ \ retrieved\ \ objects}$$

Equation 2. Definition of precision

So firstly, terms that are mentioned frequently in individual documents or extracts from a document, appear to be useful for improving recall. This suggests the use of a factor known as Term Frequency (TF) as part of a Term Weighting system, measuring the frequency of occurance of a term in a document. The TF factor has been used for Term Weighting for years in automatic indexing environments. Secondly, the TF factor solely does not ensure an acceptable retrieval. In particular, when the high frequency terms are not concentrated in specific documents, but instead are frequent in the entire set, all documents tend to be recovered, and this affects the precision factor. Thus, there is the need to introduce a new factor that favours the terms that are concentrated in only a few documents in the collection. The Inverse Document Frequency (IDF) is the factor that considers this aspect. The IDF factor is inversely proportional to the number of documents (n) to which a term is assigned in a set of documents N. A typical IDF factor is log (N / n) (Salton & Buckley, 1996). So the best index terms to identify the contents of a document are those able to distinguish certain individual documents from the rest of the set. This implies that the best terms should have high term frequencies, but low overall collection frequencies. A reasonable measure of the importance of a term can be obtained, therefore, by the product of term frequency and inverse document frequency (TF x IDF). It is usual to describe the weight of a term *i* in a document *j* as shown in Equation 3.

$$w_{ij} = tf_{ij} \times idf_j$$

Equation 3. Obtention of term weights; general formula

This formula was originally designed for the retrieval and extraction of documents. Eventually, it has also been used for the retrieval of any object in any set of accumulated knowledge, and has been revised and improved by other authors in order to obtain better results in Information Retrieval (Lee et al., 1997), (Zhao & Karypis, 2002), (Lertnattee & Theeramunkong, 2003), (Liu & Ke, 2007).

In short, term weights must be related somehow to the importance of an index term in the corresponding set of knowledge. There are two options for defining these weights:

- The evaluation of the weights by an expert in the field. This is based on his own perception of the importance of index terms. This method is simple, but it has the disadvantage of relying solely on the criterion of the engineer of knowledge, it is very subjective and is not able of being automated.
- Automated generation of weights using a set of rules. The most widely used method for Term Weighting, as said above, is the TF-IDF method. In this chapter, we propose a novel Fuzzy Logic-based Term Weighting method, which obtains better results for Information Retrieval.

To calculate the weight of a term, the TF-IDF approach considers two factors:

- TF: Frequency of occurrence of the term in the document. So tf_{ik} is the frequency of occurrence of the term T_k in document i.

- IDF: varies inversely with the number of documents n_k where the term T_k has been assigned in a set of N documents. The typical IDF factor is represented by the expression log (N / n_k + 0.01).

Introducing standardization to simplify the calculations, the formula finally obtained for the calculation of the weights is defined in Equation 4 (Liu et al., 2001)

$$W_{ik} = \frac{tf_{ik} \times \log(N / n_k + 0.01)}{\sqrt{\sum_{k=1}^{m} tf_{ik} \times \log(N / n_k + 0.01))^2}}$$

Equation 4. Obtention of term weights. Used formula.

A third factor that is commonly used is the document length normalization factor. Long documents usually have a much larger set of extracted terms than short documents. This fact makes it more likely that long documents are retrieved (Van Rijsbergen, 1979), (Salton & Buckley, 1996). The term weight obtained using a length normalization factor is given by Equation 5.

$$W_{ik} = \frac{w_{ik}}{\sqrt{\sum_{i=1}^{m} (w_i)^2}}$$

Equation 5. Obtention of term weights using a length normalization factor

In Equation 5, w_i correspond to the weights of the other components of the vector.

All Term Weighting tasks are shown in Figure 2.

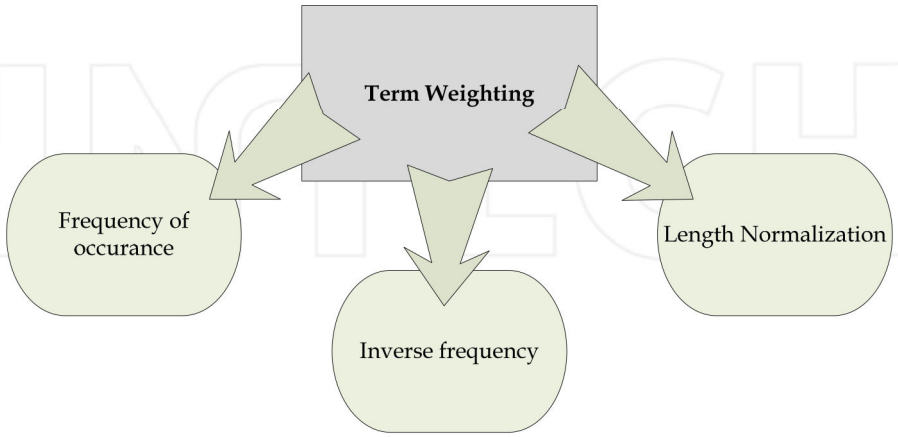


Fig. 2. Term Weighting tasks

3. Term Weighting method comparison

3.1 Term Weighting methods

The TF-IDF method works reasonably well, but has the disadvantage of not considering two aspects that we believe key:

- The first aspect is the degree of identification of the object if a determined index term is solely used in a query. This parameter has a strong influence on the final value of a weight of term if the degree of identification is high. The more a term identifies an object, a higher value has its correspondent weight. However, this parameter creates two disadvantages in terms of practical aspects when a systematic, automated Term Weighting scheme is neccessary. On the one hand, the degree of identification is not deductible from any feature of the index term, so it must be specified by the Knowlegde Engineer. The assigned values may therefore be subjective, not systematic and not univocal. On the other hand, the same index term may have a different relationship with different objects.
- The second aspect is related to the join index terms, i.e. terms that are linked to others. Join terms have lower weights as the fact that these keywords are linked is what really determines the principal object. The appearance of one of these words could refer to another object.

This chapter describes, firstly, the operation of TF-IDF method. Then, the new Term Weighting Fuzzy Logic-based method is introduced. Finally, both methods are implemented for the particular case of Information Retrieval for the University of Seville web portal, obtaining specific results of the operation of both of them. A web portal is a typical example of a disordered, vague and heterogenous set of knowledge. With this aim, an intelligent agent was designed to allow an efficient retrieval of the relevant information. This system should be valid for any set of knowledge. The system was designed to enable users to find possible answers to their queries in a set of knowledge of a great size. The whole set of knowledge was classified into different objects. These objects represent the possible answers to user queries and were organized into hierarchical groups (called Topic, Section and Object). One or more standard questions are assigned to every object and some index terms are extracted from them.

The last step is Term Weigthing; the assigned weight depends on the importance of an index term for the identification of the object. The way in which these weights are assigned is the main issue of this chapter. All the process is shown in Figure 3.

As an example of the classical TF-IDF Term Weighting method functioning, we are using the term 'library', used in the example shown in Table 1.

At Topic hierarchic level:

- 'Library' appears 6 times in Topic 6 (tf_{ik} = 6, K=6).
- 'Library' appears 10 times in other Topics (n_k = 3)
- There are 12 Topics in total (N=12) - for normalizing, it is only necessary to know the other tf_{ik} and n_k for the Topic-.
- Substituting, W_{ik} = 1.00.

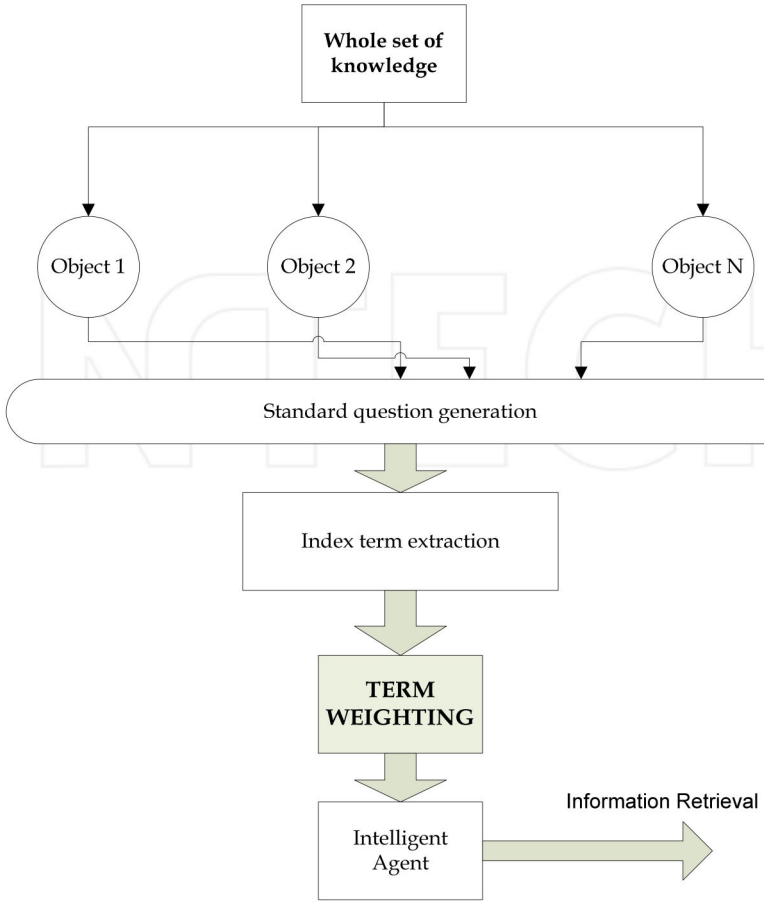


Fig. 3. Information Retrieval process.

As well, an example of the followed methodology is shown in Table 1.

STEP	EXAMPLE
Step 1: Web page identified by standard/s question/s	- Web page: http://bib.us.es/index-ides-idweb.html - Standard question: What online services are offered by the Library of the University of Seville?
Step 2: Locate standard/s question/s in the hierarchical structure.	Topic 6: Library Section 3: Online services Object 1.
Step 3: Extract index terms	Index terms: 'Library', 'services', 'online'
Step 4: Term weighting	Explained below

Table 1. Example of the followed methodology.

At Section hierarchic level:

- 'Library' appears 6 times in Section 6.3 (tf_{ik} = 6, K= 3)
- 'Library' appears 4 times in other Sections in Topic 6 (n_k = 6)
- There are 6 Sections in Topic 6 (N=6).
- Substituting, W_{ik} = 0.01. In fact, 'Library' appears in most of the Sections in Topic 6, so it is not very relevant to distinguish the desired Section inside the Topic.

At Object hierarchic level:

- 'Library' appears once in Object 6.3.1 (tf_{ik} = 1, K = 1). – Logically a term can only appear once in an Object -.
- 'Library' appears 3 times in other Topics (n_k = 3).
- There are 4 Objects in Section 6.3 (N=3).
- Substituting, W_{ik} = 0.01.

Consequently, 'Library' is relevant to find out that the Object is in Topic 6, but not very relevant to find out the definite Object, which should be found according to other terms in a user consultation.

As said above, TF-IDF has the disadvantage of not considering the degree of identification of the object if only the considered index term is used and the existance of join terms. The FL-based method provides a solution for these problems: the solution is to create a table of all the index terms and their corresponding weights for each object. This table will be created in the process of extracting the index words from the standard questions. Imprecision practically does not affect the method due to the fact that Term Weighting is based on fuzzy logic. This fact minimizes the effect of possible variations of the assigned weights.

Furthermore, the Fuzzy Logic-based method provides two important advantages:

- Term Weighting is automatic.
- The level of expertise required is much lower. Moreover, there is no need for an operator of any kind of knowledge about Fuzzy Logic, but only has to know how many times an index term appears in a certain subset and the answer to two simple questions:
- How does an index term define an object by itself?
- Are there any join terms tied to the considered index term?

For example, in the case of a website, the own web page developer may define standard questions. These questions are associated with the object - the web page -. He also should define the index for each object and answer the two questions proposed above. This greatly simplifies the process and leaves the possibility of using collaborative intelligence.

Fuzzy Logic based Term Weighting method is defined below. Four questions must be answered to determine the weight of an Index Term:

- Question 1 (Q1): How often does an index term appear in other subsets? - Related to IDF factor -.
- Question 2 (Q2): How often does an index term appear in its own subset? - Related to TF factor -.
- Question 3 (Q3): Does an index term undoubtedly define an object by itself?
- Question 4 (Q4): Is an index term joined to another one?

With the answers to these questions, a set of values is obtained. These values are the inputs to a fuzzy logic system, a Term Weight Generator. The Fuzzy Logic system output sets the weight of an index term for each hierarchical level (Figure 4).

Fig. 4. Term Weighting using Fuzzy Logic.

Next it is described how to define the system input values associated with each of the four questions (Qi). Qi are the inputs to the Fuzzy Logic system

Question 1

Term weight is partly associated to the question 'How often does an index term appear in other subsets?'. It is given by a value between 0 – if it appears many times – and 1 - if it does not appear in any other subset -. To define weights, we are considering the times that the most used terms in the whole set of knowledge appear. The list of the most used index terms is shown in Table 2.

Number of order	Index term	Number of appearances in the accumulated set of knowledge
1	Service	31
2	Services	18
3	Library	16
4	Research	15
5	Address	14
	Student	14
7	Mail	13
	Access	13
9	Electronic	12
	Computer	12
	Resources	12
12	Center	10
	Education	10
	Registration	10
	Program	10

Table 2. List of the most used words.

Provided that there are 1114 index terms defined in our case, we think that 1 % of these words must mark the border for the value 0 (11 words). Therefore, whenever an index term appears more than 12 times in other subsets, we will give it the value of 0. Associated values for every Topic are defined in Table 3.

Number of appearances	0	1	2	3	4	5	6
Associated value	1	0.9	0.8	0.7	0.64	0.59	0.53
Number of appearances	7	8	9	10	11	12	≥13
Associated value	0.47	0.41	0.36	0.3	0.2	0.1	0

Table 3. Input values associated to Q1 for topic hierarchic level.

Between 0 and 3 times appearing - approximately a third of the possible values - , we consider that an index term belongs to the so called HIGH set. Therefore, it is defined in its correspondant fuzzy set with uniformly distributed values between 0.7 and 1, as may be seen in Figure 5. Analogously, we distribute all values uniformly according to different fuzzy sets. Fuzzy sets are defined by linguistic variables LOW, MEDIUM and HIGH. Fuzzy sets are triangular, on one hand for simplicity and on the other hand because we tested other more complex types of sets (Gauss, Pi type, etc), but the results did not improve at all.

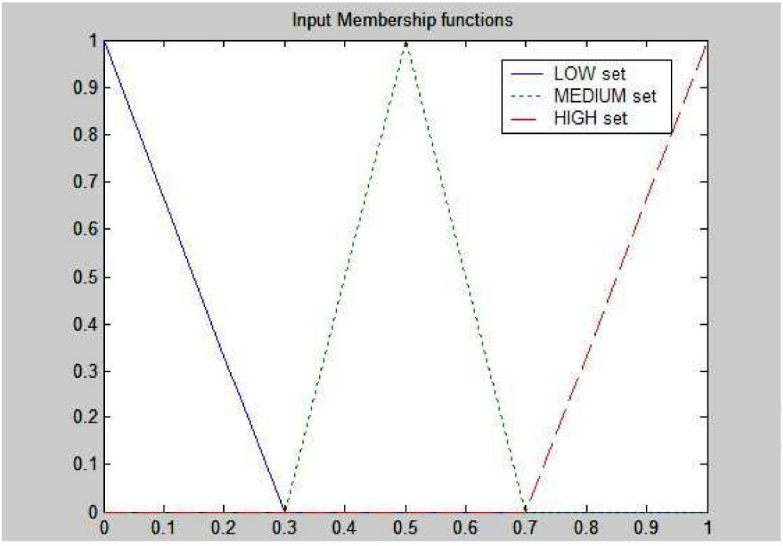


Fig. 5. Input fuzzy sets.

On the other hand, given that at each hierarchical level, a different term weight is defined, it is necessary to consider other scales to calculate the fuzzy system input values for the other hierarchical levels. As for the level of topic was considered the top level - the whole set of knowledge - , for the level of Section we consider the number occurrences of an index term on a given topic. Keeping in mind that all topics are considered, we take as reference the

value of the topic in which the index term appears more often. The process is analogous to the above described, obtaining the values shown in Table 4.

Number of appearances	0	1	2	3	4	5	≥ 6
Associated value	1	0.7	0.6	0.5	0.4	0.3	0

Table 4. Input values associated to Q1 for section hierarchic level.

To find the term weight associated with the object level, the method is slightly different. It is also based on the definition of fuzzy sets, but we do not take into account the maximum number of words per secion, but the value associated to Q1 directly passes the border between fuzzy sets when the number of objects in which it appears increases in one unit, as seen in Table 5.

Number of appearances	0	1	2	≥ 3
Associated value	1	0.7	0.3	0

Table 5. Input values associated to Q1 for object hierarchic level.

Question 2

To find the imput value to the FL system of FL with question 2, the reasoning is analogous to the one for Q1, Though, we only have to consider the frequency of occurance of an index term within a single subset of knowledge, and not the frequency of occurrence in other subsets. Logically, the more times a term appears in a subset, the greater the probability that the query is related to it. Question Q2 corresponds to the TF factor.

Looking again at the list of index terms used in a topic, we obtain the values shown in Tables 6 and 7. It has been taken into account that the more times an index term appears in a topic or section, the greater should be the input value. These tables correspond to the values for the hierarchical levels of Topic and Section, respectively.

Number of appearances	1	2	3	4	5	≥ 6
Associated value	0	0.3	0.45	0.6	0.7	1

Table 6. Input values associated to Q2 for topic hierarchic level.

Number of appearances	1	2	3	4	5	≥ 6
Associated value	0	0.3	0.45	0.6	0.7	1

Table 7. Input values associated to Q2 for section hierarchic level.

Q2 is meaningless to determine the input value for the last hierarchical level. At this level, an index term appears only once on every object.

Question 3

For Question 3, the answer is completely subjective. In this chapter, we propose the values "Yes", "Rather" and "No". Table 8, shows the input values associated with Q3. This value is independent of hierarchical level.

Answer (Does the term itself define the Object?)	Yes	Rather	No
Associated value	1	0.5	0

Table 8. Input values associated to Q3.

For example, the developer of a web page would only have to answer "Yes", "Rather" or "No" to Question 3, without complicated mathematical formulas to describe it.

Question 4

Finally, question 4 deals with the number of index terms joined to another one. If an index term is joined to another one, its weight is lower. This is due to the fact that the term must be a join term to refer to the object in question. We propose term weight values for this question in Table 9. Again, the values 0.7 and 0.3 are a consequence of considering the border between fuzzy sets.

Joined terms to an index term	0	1	2	≥ 3
Associated value	1	0.7	0.3	0

Table 9. Input values associated to Q3.

After considering all these factors, fuzzy rules must be defined. In the case of Topic and Section hierarchical levels, we must consider the four input values that are associated with questions Q1, Q2, Q3 and Q4. Four output fuzzy sets have been also defined: HIGH, MEDIUM-HIGH, MEDIUM-LOW AND LOW. For the definition of the fuzzy rules for the Term Weighting system, we have used basically the following criteria:

- A high value of Q1 (IDF-related factor) implies that the term is not very present in other sets of knowledge. In this case, the output will be high, unless the term itself has very little importance (low Q3) or it is joined to many terms (low Q4).
- A high value of Q2 (TF-related factor), usually implies a high output value, since the index term is very present in a set of knowledge. However, if Q1 has a low value means that the term is present throughout the whole set of knowledge, so it is not very useful for extracting information.
- Q3 is a very important parameter, since if one term defines itself very well to a particular object, it is much easier to find the object.
- A low value of Q4 makes an index term less important, since it is associated with other terms. This fact causes a lower output value.

The combination of the four inputs and the three input fuzzy sets provides 81 possible combinations, which are summarized in Table 10.

In the object level (the last hierarchic level), Question 2 is discarded. Therefore, there is a change in the rules, although the criteria for the definition of fuzzy rules are similar to the previous case. An input less reduces the number of rules to twenty seven.

3.2 Example of the followed methodology

An example of the followed methodology is shown below. A comparision with the classical TF-IDF is done, starting from the definition of an object in the database of the Web portal of

the University of Seville. The following example shows the difference between applying the TF-IDF method and applying the Fuzzy Logic-based one.

Rule number	Rule definition	Output
R1	IF Q1 = HIGH and Q2 ≠ LOW	At least MEDIUM-HIGH
R2	IF Q1 = MEDIUM and Q2 = HIGH	At least MEDIUM-HIGH
R3	IF Q1 = HIGH and Q2 = LOW	Depends on other Questions
R4	IF Q1 = HIGH and Q2 = LOW	Depends on other Questions
R5	IF Q3 = HIGH	At least MEDIUM-HIGH
R6	IF Q4 = LOW	Descends a level
R7	IF Q4 = MEDIUM	If the Output is MEDIUM-LOW, it descends to LOW
R8	IF (R1 and R2) or (R1 and R5) or (R2 and R5)	HIGH
R9	In any other case	MEDIUM-LOW

Table 10. Fuzzy rules.

In the Web portal database, Object 6.3.1 (http://bib.us.es/index-ides-idweb.html) is defined by the following standard question:

What online services are offered by the Library of the University of Seville?

If we consider the term 'library':

At Topic hierarchic level:

- 'Library' appears 6 times in other Topics in the whole set of knowledge, so that the value associated to Q1 is 0.53.
- 'Library' appears 10 times in Topic 6, so that the value associated to Q2 is 1.
- The response to Q3 is 'Rather' in 7 of the 10 times and 'No' in the other three, so that the value associated to Q3 is a weighted average: (7*0.5 + 3*0)/10 = 0.35.
- Term 'Library' is tied to one term 7 times and it is tied to two terms once. Therefore, the average is 1.1 terms. A linear extrapolation leads to a value associated to Q4 of 0.66.
- With all the values as inputs for the fuzzy logic engine, we obtain a term weight of 0.56.

At Section hierarchic level:

- 'Library' appears 6 times in other Sections corresponding to Topic 6, so that the value associated to Q1 is 0.
- 'Library' appears 4 times in Topic 6, so that the value associated to Q2 is 0.6.
- The response to Q3 is 'Rather' in three of the four cases, so that the value associated to Q3 is (3*0.5 + 1*0)/4 = 0.375.
- Term 'Library' is tied to one term 5 times and it is tied to two terms once so that the value associated to Q4 is 0.63.

- With all the values as inputs for the fuzzy logic engine, we obtain a term weight of 0.13.

At Object hierarchich level:

- 'Library' appears 3 times in other Objects corresponding to Section 6.3, so that the value associated to Q1 is 0.
- The response to Q3 is 'Rather', so that the value associated to Q3 is 0.5.
- Term 'Library' is tied to one term twice and it is tied to two terms once so that the value associated to Q4 is 0.57.
- With all the values as inputs for the fuzzy logic engine, we obtain a term weight of 0.33.

A summary of the values for the index term 'library' is shown in Table 11.

Hierarchic levels		Q1 value	Q2 value	Q3 value	Q4 value	Term Weight
Topic level (Topic 6)	**TF-IDF Method**	-	-	-	-	1
	Fuzzy Logic-based method	0.53	1	0.35	0.66	0.56
Section level (Section 3)	**TF-IDF Method**	-	-	-	-	0.01
	Fuzzy Logic-based method	0	0.6	0.375	0.63	0.13
Object level (Object 1)	**TF-IDF Method**	-	-	-	-	0.01
	Fuzzy Logic-based method	0	-	0.5	0.57	0.33

Table 11. Comparison of Term Weight values.

We may see the difference with the corresponding weight for the TF-IDF method - a value W_{ik} = 0.01 had been obtained), but this is just what we were looking for: not only the desired object is found, but also the ones that are more closely related to it. The word 'library' has a small weight for the TF-IDF method because it can not distinguish between the objects of Section 6.3. However, in this case all the objects will be retrieved, as they are interrelated. The weights of other terms determine the object which has a higher level of certainty.

4. Tests and results

4.1 General tests

Tests were held on the website of the University of Seville. 253 objects were defined, and grouped in a hierarchical structure, with 12 topics. Every topic has a variable number of sections and objects. From these 253 objects, 2107 standard questions were extracted. More

than half of them were not used for these tests, as they were similar to others and did not contribute much to the results. Finally, the number of standard questions used for the tests was 914. Also, several types of standard questions were defined.

Depending on the nature of the considered object, we defined different types of standard questions, such as:

- A single primary standard question, which is the one that best defines an object. This question must always be associated to every object, the others types of standard questions are optional.
- Standard questions that take into account synonyms of some of the index terms used in the main standard question (e.g., "report" as a synonym for "document").We have called them synonim standard questions.
- Standard questions that take into account that a user may search for an object, but his question may be inaccurate or may be he does not know the proper jargon (e.g., "broken table" for "repairing service"). We have called them imprecise standard questions.
- Standard questions that are related to the main question associated with the object, but are more specific (e.g., "I'd like to find some information about the curriculum of Computer Science" to "I'd like to find some information about the curriculum for the courses offered by the University of Seville "). We have called them specific standard questions.
- Standard questions created by a feedback system. Most frequent user queries may be used.

For our tests, we considered the types of standard questions shown in Table 12.

Type of standard question	**Number of questions**
Main standard questions	252
Synonim standard questions	308
Imprecise standard questions	125
Specific standard questions	229
Feedback standard questions	0
Total standard questions	**914**

Table 12. Types of standard questions.

The standard questions were used as inputs in a Fuzzy Logic-based system. The outputs of the system are the objects with a degree a certainty greater than a certain threshold. To compare results, we considered the position in which the correct answer appears among the total number of answers identified as probable.

First of all, we shall define the thresholds to overcome in the Fuzzy Logic system. Thus, topics and sections that are not related to the object to be identified are removed. This is one of the advantages of using a hierarchical structure. Processing time is better as many subsets of knowledge are discarded. Anyway, it is desirable not to discard too many objects, in order to also obtain the related ones. The ideal is to retrieve between one and five answers for the user. The results of the consultation were sorted in 5 categories:

- Category Cat1: the correct answer is retrieved as the only answer or it is the one that has a higher degree of certainty between the answers retrieved by the system.
- Category Cat2: The correct answer is retrieved between the three answers with higher degree of certainty -excluding the previous case -.
- Category Cat3: The correct answer is retrieved between the five answers with higher degree of certainty - excluding the previous cases -.
- Category Cat4: The correct answer is retrieved, but not between the five answers with higher degree of certainty.
- Category Cat5: The correct answer is not retrieved by system.

Results are shown in Table 13

Method	Cat1	Cat2	Cat3	Cat4	Cat5	Total
TF-IDF method	466 (50.98%)	223 (24.40%)	53 (5.80%)	79 (8.64%)	93 (10.18%)	914
FL-based method	710 (77.68%)	108 (11.82%)	27 (2.95%)	28 (3.06%)	41 (4.49%)	914

Table 13. Information Retrieval results of using both Term Weighting methods.

The results obtained with the TF-IDF method are quite reasonable. 81.18% of the objects are retrieved among the top 5 choices and more than half of the objects are retrieved in the first place, Fuzzy Logic-based method is clearly better. 92.45% of the objects are retrieved and more than three-quarters are retrieved in the first place.

4.2 Tests according to the type of standard questions

In order to refine the conclusions about both Term Weighting methods, it is important to make a more thorough analysis of the results. We submitted to both Term Weighting methods to a comprehensive analysis according to the type of standard question. Results are shown in the Table 14.

According to the results, the TF-IDF method works relatively well considering the number of objects retrieved. Though, the Fuzzy Logic-based method is more precise, retrieving 91.67% of the objects in the first place. On the other hand, good results for this type of questions are logical, since questions correspond to supposedly well-made user queries.

For synonymous standard questions, the conclusions are similar: the results obtained using the Fuzzy Logic-based method are better than those achieved with TF-IDF method, especially in regard to precision. Though, the TF-IDF method also ensures good results. However, queries are not precise, so the performance is worse for the TF-IDF method than it is for the Fuzzy Logic-based method. This fact gives an idea of fuzzy logic as an ideal tool for adding more flexibility to the system. Anyway, the results are quite similar to those obtained for the main standard questions. They are only slightly worse, since synonim standard questions are similar to the main standard questions.

The difference is even more noticeable in regard to imprecise standard questions and specific standard questions. Imprecise standard questions are detected nearly as well as the main standard questions in the case of Fuzzy Logic-based method. This is another reason to confirm the appropriateness of using Fuzzy Logic. As for the specific standard questions, we

Type of standard question		Cat1	Cat2	Cat3	Cat4	Cat5	Total
Main standard questions	TF-IDF Method	171 (67.86%)	58 (23.02%)	6 (2.38%)	6 (2.38%)	11 (4.37%)	252
	Fuzzy Logic-based method	231 (91.67%)	13 (5.16%)	2 (0.79%)	0 (0.00 %)	6 (2.38%)	252
Synonim standard questions	TF-IDF Method	177 (57.46%)	86 (27.92%)	13 (4.22%)	15 (4.87%)	17 (5.52%)	308
	Fuzzy Logic-based method	252 (81.82%)	41 (13.31%)	3 (0.97%)	5 (1.62%)	47 (2.27%)	308
Imprecise standard questions	TF-IDF Method	74 (59.20%)	32 (25.60%)	6 (4.80%)	1 (0.80%)	12 (9.60%)	125
	Fuzzy Logic-based method	111 (88.80%)	5 (4.00%)	0 (0.00 %)	0 (0.00 %)	9 (7.20%)	125
Specific standard questions	TF-IDF Method	46 (20.08%)	49 (21.40%)	26(11.35%)	55(24.01%)	52 (22.71%)	229
	Fuzzy Logic-based method	107 (46.72%)	53 (23.14%)	24 (10.48%)	23(10.04%)	22 (9.61%)	229

Table 14. Information Retrieval results of using both Term Weighting methods, according to the type of standard question.

get the worst result by far among all classes of standard questions. This is a logical fact, considering that these questions are associated with the main standard question, but it is more concrete. In fact, it is usual for such specific questions to belong to a list within a whole. This way, there may be objects that are more related to the query than the requiered object itself. This is hardly a drawback, since both objects are retrieved to the user - the more specific one and the more general one -. The own user must choose which one is the most accurate. This case shows more clearly that the fact of using Fuzzy Logic allows the user to extract a larger number of objects.

4.3 Tests according to the number of standard questions

Another aspect to consider in the analysis of the results is the number of standard questions assigned to every object. Obviously, an object that is well defined by a single standard question is very specific. Thus, it is easy to extract the object from the complete set of knowledge. However, there are objects that contain very vague or imprecise information, making it necessary to define several standard questions for every object. For this study, the objects are grouped into the following:

- Group 1: the object is defined by a single standard question.
- Group 2: the object is defined by two to five standard questions.
- Group 3: the object is defined by six to ten standard questions.
- Group 4: the object is defined by more than ten standard questions.

Obviously, groups 1 and 2 are more numerous, since it is less common that many questions have the same response. However, the objects from the groups 3 and 4 correspond to a wide range of standard questions, so they are equally important. In Table 15 the number of objects for each of these groups is defined.

Group number	Number of standard questions per object	Number of objects
Group 1	1	95
Group 2	2-5	108
Group 3	6-10	22
Group 4	> 10	28

Table 15. Groups according to the number of standard questions per object.

To analyze the results, the position in which the required object is retrieved must be considered. We consider the retrieval of most of the standard questions that define that object. For example, if an object is defined by 15 standard questions and, for 10 of them, the object is retrieved in second place, it is considered that the object has actually been retrieved in second place.

In short, this study does not focus on the answers to standard questions, but on the correctly retrieved objects. This provides a new element for the system analysis. Results are shown in Table 16.

For group 1, the results are almost perfect for the Fuzzy Logic-based method, as nearly all the objects are retrieved in the first place (about 94%). However, the TF-IDF method, though not as accurate, resists the comparison. This behaviour is repeated in group 2. The objects are often retrieved by both methods among the top three items. Though, the Fuzzy Logic-based method is better for its accuracy, retrieving over 92% of the objects in the first place. In view of the tests, we conclude that the results are very good for both methods when up to five standard questions are defined. Although the results are better for the novel Fuzzy Logic-based Term Weighting method, they are also quite reasonable for the classical TF-IDF Term Weighting method.

However, the largest advantage of using Fuzzy Logic for Term Weighting occurs when many standard questions per object are defined, i.e. when the information is confusing, disordered or imprecise. For the case of group 3, where objects are defined by among six and ten standard questions per object type, we observe that there is a significant difference between the TF-IDF classical method and the proposed Fuzzy Logic-based method. Although both methods retrieve all the objects, there is a big difference in the way they are retrieved, especially on the accuracy of the information extraction. 86% of the objects are retrieved in first place using the Fuzzy Logic-based method, while only 45% using the TF-IDF classical method.

Type of standard question		Cat1	Cat2	Cat3	Cat4	Cat5	Total
Group 1	TF-IDF Method	74 (77.89%)	16 (16.84%)	1 (1.05%)	1 (1.05%)	3 (3.16%)	95
	Fuzzy Logic-based method	89 (93.68%)	3 (3.16%)	2 (2.10%)	0 (0.00 %)	1 (1.05%)	95
Group 2	TF-IDF Method	86 (79.63%)	21 (19.44%)	1 (0.93%)	0 (0.00 %)	0 (0.00 %)	108
	Fuzzy Logic-based method	100 (92.59%)	7 (6.48%)	0 (0.00 %)	0 (0.00 %)	1 (0.93%)	108
Group 3	TF-IDF Method	10 (45.45%)	9 (40.91%)	3 (13.63%)	0 (0.00 %)	0 (0.00 %)	22
	Fuzzy Logic-based method	19 (86.36%)	3 (13.63%)	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)	22
Group 4	TF-IDF Method	10 (35.71%)	10 (35.71%)	3 (10.71%)	2 (7.14%)	3 (10.71%)	28
	Fuzzy Logic-based method	21 (75.00%)	4 (14.29%)	1 (3.57%)	1 (3.57%)	1 (3.57%)	28

Table 16. Information Retrieval results of using both Term Weighting methods, according to the number of standard questions per object.

The difference is even more marked when more than ten standard questions per object are defined. In this case, it is obvious that none of the questions clearly define the object, so that information is clearly vague. While using the Fuzzy Logic-based method, more than 96% of the objects are retrieved - with 75% of them in the first place -, with the TF-IDF method correctly, only 82% of the objects are retrieved. Furthermore, only 35.7% of these objects are extracted in the first place.

In view of the table, we observe that the more standard questions per object, the better the results of the Fuzzy Logic-based method, compared with those obtained with the classical TF-IDF method. Therefore, the obvious conclusion is that the more convoluted, messy and confusing is the information, the better the Fuzzy Logic-based Term Weighting method is compared to the classical one. This makes Fuzzy Logic-based Term Weighting an ideal tool for the case of information extraction in a web portal.

5. Future research directions

We suggest the application of other Computational Intelligence techniques apart from Fuzzy Logic for Term Weighting. Among these techniques, we believe that the so-called

neuro-fuzzy techniques represent a very interesting field, as they combine human reasoning provided by Fuzzy Logic and the connection-based structure of Artificial Neural Networks, taking advantage of both techniques. One possible application is the creation of fuzzy rules by means of an Artificial Neural Network system.

Another possible future direction is to check the validity of this method in other environments containing inaccurate, vague and heterogeneous data.

6. Conclusion

The difficulty to distinguish the necessary information from the huge quantity of unnecessary data has enhanced the use of Information Retrieval recently. Especially, the so-called Vector Space Model is much extended. Vector Space Model is based on the use of index terms. These index terms are associated with certain weights, which represent the importance of these terms in the considered set of knowledge. In this chapter, we propose the development of a novel automatic Fuzzy Logic-based Term Weighting method for Vector Space Model. This method improves the TF-IDF Term Weighting classic method for its flexibility. The use of Fuzzy Logic is very appropiate in heterogeneous, vague, imprecise, or not in order information environments.

Fuzzy Logic-based method is similar to TF-IDF, but also considers two aspects that the TF-IDF does not: the degree of identification of the object if a determined index term is solely used in a query; and the existance of join index terms. Term Weighting is automatic. The level of expertise required is low, so there is no need for an operator of any kind of knowledge about Fuzzy Logic. Therefore, an operator only has to know how many times an index term appears in a certain subset and the answer to two simple questions.

Although the results obtained with the TF-IDF method are quite reasonable, Fuzzy Logic-based method is clearly superior. Especially when user queries are not equal to the standard query or they are imprecise, we observe that the performance declines more for the TF-IDF method than for the Fuzzy Logic-based method. This fact gives us an idea of how suitable is the use of Fuzzy Logic to add more flexibility to an Information Retrieval system.

7. References

Lertnattee, V. & Theeramunkong, T. (2003). Combining homogenous classifiers for centroid-based text classification. *Proceedings of the 7th International Symposium on Computers and Communications*, pp. 1034-1039.

Lee, D.L., Chuang, H., Seamons, K., 1997. *Document ranking and the vector-space model*. IEEE Software, Vol. 14, Issue 2, pp. 67 – 75.

Liu, S., Dong, M., Zhang, H., Li, R. & Shi, Z. (2001). An approach of multi-hierarchy text classification. *Proceedings of the International Conferences on Info-tech and Info-net*. Beijing. Vol 3, pp. 95 – 100.

Raghavan, V.V. & Wong, S.K. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, Vol.37 (5), p. 279-87.

Ruiz, M. & Srinivasan, P. (1998). Automatic Text Categorization Using Neural Networks. *Advances in Classification Research vol. 8: Proceedings of the 8th ASIS SIG/CR*

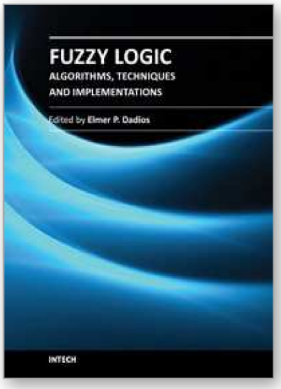
Classification Research Workshop. Ed. Efthimis Efthimiadis. Information Today, Medford:New Jersey, pp 59-72.

Salton, G. (1988). *Automatic Text Processing*. Addison-Wesley Publishing Company.

Salton, G. & Buckley, C. (1996). Term Weighting Approaches in Automatic Text Retrieval. *Technical Report TR87-881, Department of Computer Science, Cornell University, 1987. Information Processing and Management* Vol.32 (4), pp. 431-443.

Van Rijsbergen, C.J. (1979). *Information retrieval.* Butterworths.

Zhao, Y. & Karypis, G. (2002). Improving precategorized collection retrieval by using supervised term weighting schemes. *Proceedings of the International Conference on Information Technology: Coding and Computing*, pp 16 – 21.



Fuzzy Logic - Algorithms, Techniques and Implementations
Edited by Prof. Elmer Dadios

ISBN 978-953-51-0393-6
Hard cover, 294 pages
Publisher InTech
Published online 28, March, 2012
Published in print edition March, 2012

Fuzzy Logic is becoming an essential method of solving problems in all domains. It gives tremendous impact on the design of autonomous intelligent systems. The purpose of this book is to introduce Hybrid Algorithms, Techniques, and Implementations of Fuzzy Logic. The book consists of thirteen chapters highlighting models and principles of fuzzy logic and issues on its techniques and implementations. The intended readers of this book are engineers, researchers, and graduate students interested in fuzzy logic systems.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821